DOCUMENT RESUME

ED 271 086 HE 019 469

TITLE	Computer Designed Instruction & Testing.
INSTITUTION	New Mexico State Univ., Las Cruces.
PUB DATE	Jan 85
NOTE	241p.; Contains 13 papers on this theme from: "Teaching to Potential." Proceedings of the Regional Conference on University Teaching, (1st, Las Cruces, NM, January 9-10, 1985), see HE 019 471.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC10 Plus Postage.
DESCRIPTORS	Authoring Aids (Programing); Business Administration Education; Case Studies; *College Instruction; College Mathematics; College Science; *Computer Assisted Instruction; *Computer Assisted Testing; *Courseware; Educational Testing; *Feedback; Higher Education; Microcomputers; Self Evaluation (Individuals); Statistics; Teacher Education

ABSTRACT

Research findings on computer designed instruction and testing at the college level are discussed in 13 papers from the first Regional Conference on University Teaching at New Mexico State University. Titles and authors are as follows: "Don't Bother Me with Instructional Design, I'm Busy Programming! Suggestions for More Effective Educational Software" (M. David Merrill); "Computerized Exam Administering System and Student Gradebook" (Greg Baker, Bob Creel); "A Retesting on Demand Strategy for General Physics Using Computer Generated Problems" (A. F. Burr); "Computer Based Special Education Teacher Training" (G. Phillip Cartwright); "Some Uses of the Computer in Teaching, Training and Testing: Student/Learner Self-Assessment Responding" (Darwin Hunt); "A Computer on Every Desk; Implications for the Educational Process" (Kenneth L. Krause, William E. Richardson, Lawrence G. Jones); "Writing CAI Tutorials with PILOT" (Robert Lucking); "Computerized Testing in Introductory Business Statistics: A Case Study" (Paul H. Randolph, W. J. Conover); "Creating Cost Effective CAI for the College Classroom: The Role of Templated Authoring Aids" (Boyd Richards); "Utilizing the Microcomputer as a Tax Planning Tool in Business and Education" (Stanley S. Scott); "Some Current and Projected Uses of Computer Technology in the Teaching of Statistics" (N. Scott Urquhart); "Students, Learning, and Computers: Surviving the Changes" (James Wade); and "Computer Aided Instruction for Freshman Chemistry at the U.S. Air Force Academy" (James R. Wright, Larry D. Strawser, John R. Balyeat, Larry P. Davis, Jesse L. James, Phillip Jung, Jonathan C. Noetzel, Donn M. Storch). (SW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED271086

HE019469

THEME: COMPUTER DESIGNED INSTRUCTION & TESTING

DON'T BOTHER ME WITH INSTRUCTIONAL DESIGN I'M BUSY PROGRAMMING!

Suggestions for more effective educational software

by

M. David Merrill

University of Southern California

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Committee on College Teaching, New Mexico State University
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☑ Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

THEME: Computer Designed Instruction & Testing

Page

M. David Merrill - Speaker - Don't Bother Me With Instructional Design, I'm Busy Programming! Suggestions For More Effective Educational Software 238

Greg Baker, Bob Creel - Computerized Exam Administering System and Student Gradebook 282

A. F. Burr - A Retesting on Demand Strategy for General Physics Using Computer Generated Problems 300

G. Phillip Cartwright - Computer Based Special Education Teacher Training 320

Darwin Hunt - Some Uses of the Computer in Teaching, Training and Testing: Student/Learner Self-Assessment Responding 324

Kenneth L. Krause, William E. Richardson, Lawrence G. Jones A Computer on Every Desk; Implications for the Educational Process 340

Robert Lucking - Writing CAI Tutorials With PILOT 368

Paul H. Randolph, W. J. Conover - Computerized Testing in Introductory Business Statistics: A Case Study 380

Boyd Richards - Creating Cost Effective CAI for the College Classroom: The Role of Templated Authoring Aids 390

Stanley S. Scott - Utilizing the Microcomputer as a Tax Planning Tool in Business and Education 408

N. Scott Urquhart - Some Current and Projected Uses of Computer Technology in the Teaching of Statistics 422

James Wade - Students, Learning and Computers: Surviving the Changes 442

James R. Wright, Larry D. Strawser, John R. Balyeat, Larry P. Davis, Jesse L. James, Phillip Jung, Jonathan C. Noetzel, Donn M. Storch - Computer Aided Instruction for Freshman Chemistry at the U. S. Air Force Academy 464

BEST COPY AVAILABLE

DON'T BOTHER ME WITH INSTRUCTIONAL DESIGN I'M BUSY PROGRAMMING !

Suggestions for more effective educational software.

M. DAVID MERRILL
Professor of Educational Psychology and Technology
University of Southern California

A recent comment by Larry Lipsitz, the editor of Educational Technology Magazine, caused me to remove my PASCAL disk from my disk drive, to temporarily put aside my attempts at programming educational software and to boot my word processor and write a few comments about educational program design. "We publish very few instructional design papers anymore;" he said, "I think instructional design is dead!"

"Dead!" I exclaimed. "How can instructional design be dead? Hundreds of people, most of whom have never even heard of instructional design, are at this very moment sitting in front of their personal computers entering computer code to create educational software. Many of these programs have been, are being or will be reproduced and sold to unsuspecting parents, teachers and students. If these self-styled cottage-industry educational technologists are not engaged in instructional design what are they doing?"

Rather than being dead instructional design is undergoing a new birth. The academic approach of formal education is being challenged by creative ideas that are radically different from the past. The explosion of new software promises greater achievement than has ever before been possible.

However, for every effective innovative program that appears there also appears many programs that are innovative only because they are on the computer. These products frequently lack even the fundamentals of effective educational design. In this paper we will explore a few basic techniques of instructional design which are too often ignored by the developers of many of these new educational products.

The earliest educational programs were merely drill and practice and were little more than workbooks put on the computer. In fact some publishers did exactly that, put their already published workbooks on the computer. The result is a very dull program that may provide some skill development if the student already knows the material and if sufficient external reward or punishment can be arranged to keep the student at what otherwise is a very boring activity. Unfortunately many people still equate computer based education with such boring drill and practice programs. Advocates often cite the patience of the computer as a virtue of such programs. Little is said about the patience of the student.

Figure 1 illustrates a short sequence of such a program from a major publisher. There is no instruction prior to what is illustrated. After signing on, the student is presented frame A. The ? symbol is flashing. The student responds by typing in the missing number. If the student is incorrect frame B appears. The incorrect response flashes. After two incorrect tries the student is given the correct answer and presented the next problem. If the student is correct frame C appears. The cute graphic apple with the worm is in color and the worm crawls out of the apple. Pressing the return key gets the next problem. There are only three correct reward graphics. Frame D shows a dragon which breaths fire when the student is correct. The effectiveness of such reinforcers is quickly dissipated and the student is left with the boring task of doing one problem after another with nothing but right and wrong feedback. If the student does not know how to do the problem there is no remedial help, except to call the teacher. Ironically, this program is very expensive!

Too many educational product designers assume that their only responsibility is to provide opportunities for practice. In this author's opinion educational software should do more than merely present a workbook to the student. With the capacity provided by a personal computer courseware should provide not only opportunity for practice but information about how to perform the task as well. Adequate educational software should teach.

In this paper we shall review three classes of instructional design techniques that are necessary if educational courseware is to teach. These techniques can be grouped into those that affect three areas of instructional design: instructional strategy, screen design and human factors. Each of the techniques discussed will be illustrated by two versions of a program on poetic meter. The NOT-SO-GOOD version appears in figures 2 and 3. This is really a straw man program created especially to illustrate what not to do. Perhaps it resembles programs you have used. The second version (figures 5 through 11) is a commercial product designed by the author. In this product we have attempted to use most of the design techniques discussed. The displays included represent only a few of the displays in the program.

INSTRUCTIONAL STRATEGIES

This section identifies design strategies which are believed to have a direct impact on the amount of learning that occurs as a result of instruction. Instructional design strategy involves the arrangement of content elements and other information in such a way that learning is facilitated. In this section we have attempted to identify a few strategies which have been shown to enhance or inhibit learning.

Page Turning.

In the early days of educational television lectures were often put on the TV screen. The result was deadly. Today we have textbooks put on the monitor. Also deadly! If the monitor merely substitutes for a page of text there is little reason to put the material on the computer. Too many educational computer programs present a paragraph of text and then asked the student to press the space-bar or return key to continue. Continuing means to turn the page and see the next paragraph of text.

Many authors and instructors are accustomed to one-way linear presentations such as lectures and textbooks. In such presentations student interaction is seen as an interruption not an essential part of the presentation. As a consequence few designers conceptualize instruction as a conversation or if they do the input from the student seems to be of secondary importance serving only to indicate readiness to continue.

Look at figure 2. This represents the first few displays of the NOT-SO-GOOD poetic meter program. Frames A, B and C are consecutive; several displays are skipped between frames C and D. This is an example of the text-on-the-screen characteristic of too many educational software packages. A paragraph is presented and then the computer waits until the student presses the return key which results in another paragraph being presented. The material takes no advantage of the dynamic nature of the screen. As can be seen from the illustrations one paragraph scrolls up to make room for the next at the bottom of the screen.

To be effective educational software must be more than text on the monitor. There is no advantage to having students read text from a monitor which they can more easily read from the printed page. In short, **avoid merely putting text on the screen; avoid mere page turning.**

Generality-rich, example-poor.

Since, as an instructor, we usually understand what we are presenting there is a tendency to present one idea after another with little illustration. Illustrations and examples often seem to slow down the presentation. Text books often contain few illustrations because of the need to cover the subject matter in limited space. Adding unnecessary illustrative material "pads" the book making it unnecessarily lengthy and expensive. As a consequence of this disposition much of the CAI which has been prepared models lectures and textbooks rather than conversations. Ideas are presented one after another with little illustration and little opportunity for student interaction. As a consequence students do not have a chance to digest one idea before the next one comes.

Look at figure 2. Note that one idea tumbles after another each time the return key is pressed. In frame A the purpose of the lesson is stated but not illustrated. In frame B important prerequisite information is reviewed but not illustrated. In frame C three different ideas, stressed syllables, rhythm and poetic feet are each presented one after another without illustration.

Instruction is more effective if each major idea is represented by a general statement of the idea (generality) followed by examples and finally by the opportunity for the student to demonstrate that the idea is understood (practice). In short, **avoid generality-rich but example-poor presentations. Each idea should be presented by a generality, examples and practice.**

Remember Only.

Perhaps the biggest problem in all education is the tendency to tell the student something and then ask the student to repeat what was said rather than applying the idea to a new situation. It is easy to ask a student to repeat what has been learned but much more difficult to figure out how the student should apply the information. Application means that we must find specific cases which can be represented in our instructional materials and then we must figure out what the student does when he applies the information that has been presented. Furthermore all of us have had many examples of remember testing but many fewer examples of application testing. In formal educational settings we are accustomed to recall but not to application.

In figure 3, frames B and C, we see examples of remember questions. In these displays the student is asked to remember the definitions of each kind of meter. If the student is right the next question is presented; if wrong a message instructs the student to try again. The try again message is repeated until the student stumbles onto the correct answer. Too often this is the only kind of practice that is provided for the student. However, in the NOT-SO-GOOD Poetic Meter program the student is also ask to apply the definitions by identifying the type of meter involved in a series of words (See figure 3, frame D). A running total indicates how many the student correctly classified.

Too often educational products neglect having the student apply what has been learned. Or if there is application it is insufficient to allow the student to demonstrate what has been learned or for the designer to determine if the student has learned what was taught. Instruction is more effective if the student is required to apply ideas to new situations rather than merely repeating the statement of the idea. In short, **avoid remember only practice.**

Attention focusing.

Not all communication is designed to be instructional. One of the things that characterizes instructional communication is the addition of information that is not part of the subject matter but which is included to help the student understand the subject matter content. Text books use headings, bold face lettering, italics, arrows, exploded drawings and many other devices to facilitate the student's understanding of the content presented. This attention focusing information should serve two primary functions: first, it attempts to relate the illustrations to the generalities being taught and second, it points out the critical characteristics of the example being presented.

Computer displays have a wide variety of display characteristics which can be used for this attention focusing function. All of the characteristics of text presentation are also present on computer displays. For example, words can be underlined, appear in bold face, italic or other type styles. The display can include color which can be used to tie ideas together or emphasize important portions of the display. In addition, however, there are many dynamic characteristics of the computer display that are not available in print. These include: timing where text is put up a little at a time for emphasis, animation where text or figures are moved about to show relationships, flashing or inverse text which can direct attention to a given portion of the screen, sound which can attract attention or show patterns in the presentation.

Too often these characteristics are used for entertainment and have no instructional function. Putting text up letter by letter, word by word or moving across the screen like a sign may be clever but often only serves to make the text harder to read. Having the computer beep when text is entered or displayed often does little more than disturb others in the room. Colorful displays that show rainbow patterns of text or figures too often are only aesthetic and distract from the purpose of the instruction. If these devices are used they should serve an instructional purpose. If they do not they may prove to be more of a distraction than a facilitation.

Look at figure 1 again. The animated figures used for reinforcement are entertaining the first time they appear but they have no instructional value over that of saying, "You are right". They do not show the relationship between an example and a generality nor do they focus the students attention on relevant aspects of the examples.

Look at figure 4. This is a short sequence from another commercial program on fractions. (Edu-Ware) In frame A an animated figure pulls the shading from the first box across the screen to the second box (frames B and C). After moving the shading the figure walks to the bottom of the screen (frame D) to point to the symbolic equivalent of what has been done. This enables the student to visually and dynamically see that 1/2 is equal to 2/4. In this sequence of frames the animation directs the students attention to relevant characteristics of the display and helps the student understand the relationships involved. The animation is not only relevant but probably shows relationships that would be difficult to explain with words or symbols alone.

Look at figure 7, frame A. The directions at the top of the display indicate that the student should, "Listen to this passage." The syllables of the poem are then displayed one by one. Each syllable is accompanied by a tone. Stressed syllables are accompanied by a high tone whereas unstressed syllables are accompanied by a low tone. In this way the student can hear the stress pattern in the passage while seeing the passage displayed. Again the sound is used in an instructionally relevant way to direct the student's attention to the critical characteristic of stress.

The computer provides a considerable arsenal of display possibilities. If these devices are used to focus the student's attention on critical aspects of the subject matter being presented then they can substantially facilitate the learning which occurs. If, however, they are used indiscriminately merely to be clever without a clear instructional intent they may distract from, rather than enhance, the learning which may occur. In short, use **attention focusing devices to relate examples to generalities and to point out critical characteristics of the illustrative material.**

Active Mental Processing.

In an attempt to promote interactive instruction as opposed to linear presentations the early advocates of programmed instruction recommended that students should be required to respond overtly to the material being presented. Attempts to demonstrate the effectiveness of overt responding often were disappointing failing to show a distinct advantage for programs which required overt responses versus those which did not. More recent research has shown that it is not just overt responding that is important but the degree to which the student is required to mentally process the information which is being presented. Effective instructional conversations must engage the student in a way that requires thinking about the important aspects of what is being presented rather than passively observing information on the screen.

One of the techniques which promotes active mental processing is the use of rhetorical questions. A rhetorical question is one which is used to emphasize a point or introduce a topic and for which no answer is really expected. The advantage of using rhetorical questions rather than statements is that the student is more likely to engage his or her mental processes in formulating an answer rather than passively processing a statement. Then when this answer is confirmed by the presentation the point is more likely to be remembered.

Compare figure 2, frame B, with figure 6, frame A. The NOT-SO-GOOD program makes a simple declarative statement about the role of rhythm in distinguishing prose from poetry. The other program makes this same point but does it with a series of rhetorical questions ending with a rhetorical question which requires the student to actively engage in the conversation. The program does not wait for an answer to the first two questions:"What makes a poem a poem?" "Why is a poem different from prose?" The purpose of these questions is to cause the student to think about what they already know thus activating their mental processes so that when the point is made that one of the primary differences is the rhythm pattern they are anticipating this information and thus more likely to remember the point.

A related technique for promoting active mental processing is to engage the student in a conversation about the topic. Many educational programs use multiple choice questions as a way of interacting with the student. From a programming standpoint such questions are much easier to evaluate. However, in a conversation we are not interested in evaluating the students answer anyway. The questions we ask or the response we prompt are another form of rhetorical question. We request an answer but the purpose is not to see if the student knows what we have said or is able to give the correct answer but rather the purpose is to cause the student to think. If the comment made by the student is something we anticipated we want to acknowledge that the student is thinking consistent with our intent. If on the other hand the student provides some statement that we have not anticipated we carry on much like we do in live conversations when we do not understand, that is, we merely restate our point with the hope that the student will then see where we were going. We do not say, "That is wrong!" or "I do not agree." We merely continue with the conversation.

Again compare figure 2, frame B, with figure 6, frames A through D. In figure 6 frame A we anticipate that the student will make a response to the third request and the program provides a cursor to indicate to the student that we anticipate such a response. The program is written such that the student can type anything as a response. If the student is following our conversation, and is serious about interacting with the program, it is very probable that the response will be rhyme, since that is the most widely known characteristic of poetry. The program anticipates this response, including a dozen different ways to spell the word (rime, rhime, etc.). In response to this input from the student the program displays the information in frame B by acknowledging the students input and suggesting that there is yet another characteristic which we are thinking about. If the student anticipates our intent and responds with meter, beat or rhythm (including likely misspellings) the program responds with frame C. If the student should respond with meter or rhythm to the first question the program displays frame C instead of frame B. Finally the program displays frame D which now makes the point in a declarative statement. If the response to either the question on frame A or frame B is unanticipated the program responds with frame D. The result of this conversational programming is that the student has the feeling that the computer is really carrying on a conversation rather than merely presenting information. Furthermore, the student is being required to think about the idea being presented.

Other examples of this conversational approach can be seen in figure 7, frames A. In this frame after presenting an example of poetry and an example of prose the student is asked to observe the difference. Our conversation anticipates that the student will observe the regular pattern of stress or meter. The program anticipates a wide range of key words expressing this idea. If the student uses any of these key words in his or her response the program acknowledges with a simple, "Yes, passage 1 has regular or steady rhythm." If the student does not anticipate a key word then the "yes" is dropped from our reply but the rest of the statement is presented. The program then asks the student about passage 2. The conversation continues in this same way.

Figure 7, frame B provides yet another example of conversational programming. The program asks the student, "What is a stressed syllable?" Many students reply, "One which is stressed." This response does not go beyond the question so the program asks, "Can you think of another word?" If the student anticipates loud, emphasized, higher or synonyms of these words the program acknowledges with, "GOOD!" and the statement indicated on frame B. It should be noted that all of the displays are dynamic and the pieces shown come up one at a time so the second part of this display showing the passage of poetry actually appears after the student has been asked to read the passage aloud. Then the passage is chanted with sound enhancement to allow the student to compare his reading of the passage with the programs version of stressing the syllables.

Educational software should be conceptualized as a conversation rather than a lecture or a book. The interaction with the student is far more than merely asking the student if he or she is ready to continue. The interaction should not be limited to asking the students questions for the purpose of seeing if they know the "right" answer. Rather the purpose should be to stimulate thought and actively engage the student in an interchange which will lead the student to understand what we, as an author, are trying to teach. In short, **promote active mental processing by asking rhetorical questions and engaging the student in a conversation which requires constructed (as opposed to multiple choice) responses to which we do not provide right-wrong feedback but rather an anticipation of a reasonable reaction much like that which would be expected from a live conversation.**

Examples.

All of us know the cliche that there is nothing that teaches like a good example. For some reason the cliche has not found its way into the rules for effective educational software. The most common approach for tutorial programs is to immediately begin to ask the student questions, not of the rhetorical type discussed above, but questions testing the content which is being taught. If the student is uncertain about the content this approach often causes anxiety because the student does not know how to answer the question and being required to respond with the "right" answer when a student knows that he or she does not know causes frustration. Most educational software would be improved if the ideas were illustrated for the students by means of expository examples which show the application of the ideas being presented. Then, after the student has had an opportunity to study these worked examples, practice in applying the ideas is less threatening and more likely to be productive in promoting acquisition of the necessary skills.

The NOT-SO-GOOD poetry program moves immediately from a page turning presentation of ideas to a series of questions asking the student to remember these ideas and then to a series of questions asking the student to apply these definitions to new words without seeing any examples prior to this practice exercise. This is not atypical of many educational software packages. In the improved version of poetry figure 8, frame A shows one form of example presentation. The student has the opportunity to examine as many words of each type of meter as he or she feels is necessary to understand each of the meter types. Pressing the space-bar moves the pointer from one box to another. Pressing the return key causes the example word or phrase to appear in the box together with sound enhancement showing the stress pattern.

Figure 9, frame A shows a similar display for practice. After the student has seen as many examples as the student feels are necessary he or she can request the practice display by pressing the escape key. In this practice display words are presented one by one. The student uses the space-bar to point to the correct meter type. This is a form of multiple choice question. Note that in practice the intent is to determine if the student has mastered the concepts being taught. This display differs significantly from the conversational displays that were used to teach the ideas in the first place. If the student answers wrong the display indicates that the student is wrong, points to the correct answer and chants the word or phrase.

A similar set of example and practice displays are used for whole verses. The student is shown a passage of poetry. The dynamic display then shows the student how to divide each word into syllables, which syllables are stressed, where the poetic feet are divided, how the poem is chanted and finally the type of meter involved. After examining as many of these expository examples as the student feels is needed to understand each type of meter a series of practice verses are presented to the student. After the student has classified each verse the option is given to do a detailed analysis of the verse.

Too often educational software moves immediately from the presentation to the practice without presenting illustrative examples of the ideas being taught. Students are able to learn more efficiently when they have the opportunity to study examples of the ideas prior to being asked to practice applying the ideas to new examples. In short, **provide expository examples as well as practice.**

SCREEN DISPLAY

This section identifies screen display techniques which are believed to facilitate the students ability to interact with the materials. Screen display refers to the way that information is exhibited on the monitor. In this section we have attempted to identify a few display characteristics which may enhance or inhibit learning.

Scrolling.

One of the advances in computer technology was the ability of a computer display to scroll material automatically as the text being entered reached the bottom of the screen. For many applications such as programming or word processing this automatic scroll is a real advantage which eases the users task in entering material into the computer. However, what is an advantage for some applications is a disadvantage for others. When a student is trying to read a display it is very distracting if the text jumps up to the next line or worse jumps up several lines. The student loses his or her place and must then reread the material to find where they were. This interruption causes a distraction which often interferes with learning.

Students often use the location on the page as a cue to what was read. When reviewing material we often remember where it was on the page and use this spatial information to help us remember the content. When a page of text scrolls the material appears at a different place on the screen each time it is seen thus eliminating this spacial cue which often assists learning.

Look at figure 2 again. The displays shown here illustrate scrolling. When the return key is pressed the material on the screen jumps up line by line as the new material is being written at the bottom of the screen. The student must read the text as it jumps or wait until the whole message is written before reading the screen. Also when the scrolling stops what was just read now appears in whole or in part at the top of the screen and the new material appears at the bottom of the screen. The student may be confused about what has been read and what is new.

While scrolling is an advantage for word processing, programming and other applications it is usually a disadvantage for educational programs. There may be some situations where it has an instructional purpose but usually it is an unnecessary distraction and should be avoided. In short, **no scrolling for educational programs.**

Text Output.

In a text book the page is already filled with text when the student opens the book. With a computer display the text is written on the display before the student's eyes. Unimaginative programs often model the book and place a whole screen full of text up at once so the monitor resembles a page of text. Merely placing text on the screen ignores the dynamic character of the computer screen and the possibilities for using text output to enhance the instructional effectiveness of the presentation.

One of the dangers of the dynamic display of text stems from the fact that students read at vastly different rates. When the text is removed from the screen under program control there is a danger that the student has not yet finished reading the message. This is like reading a book with someone who reads faster and having them turn the page when you are still in the middle. When the text is removed before the student has finished reading the resulting confusion and frustration considerably inhibits learning. It is a good idea to allow the student to indicate when he or she is ready to continue (a good aspect of the "please press return to continue" message). Or, when text is removed by the program there should be some means for the student to retrieve that which was removed.

Look at figure 7, frame D. This is a dynamic display which presents the information a little at a time with a slight delay between each new part. The title comes up first, "FOUR KINDS OF POETIC FEET", followed by a 2 second delay. Then the four boxes appear. The headings are written one by one: "first syllable stressed", delay, "last syllable stressed", delay, "two syllables", delay, "three syllables", delay. Then the names of the four types of meter appear one by one: "TROCHAIC", delay, "IAMBIC", delay, etc. Then each type of meter is illustrated. The name is repeated, "TROCHAIC" with the definition, "2 syllables, 1st stressed". Then the direction, "LISTEN" flashes a few times along with the arrow pointing to the lower box. Finally an example word is placed in the box one syllable at a time while the stress is indicated by sound. After a delay of a few seconds the next type of meter is shown in the box until each of the four types have been illustrated. Examples of the different types of meter are presented and removed under program control. What if the student has not finished studying one type before the next appears? After all four types of meter have been illustrated a black stripe appears at the bottom of the display like the one shown in frame B. The directions, "[RET] REPEATS" means that pressing the return key will repeat the entire dynamic display again. In this way the student can control the text output even though individual elements are under program control.

For very short messages it is not necessary to provide means for the student to repeat the text presentation. For example, look at figure 9, frame B. When the student chooses the wrong type of meter the program displays the message, "WRONG" and shows an arrow pointing to the correct choice. Then the program chants the correct version of the word. The program then goes to the next word. The feedback information is presented under program control. In this case allowing the student to tell when to remove the "WRONG" and go on requires an unnecessary response on the part of the student since there is no difficulty in understanding the feedback message.

Computer displays, unlike text, can use the dynamic presentation of text and graphic material to focus the student's attention and facilitate learning. However, if critical text is removed before the student has had a chance to finish studying the information the result is frustration and interference with learning. A means should be provided for the student to indicate when he or she is ready to continue or for repeating information which is presented under program control. In short, **the student should control text output. Never erase critical information until the student indicates readiness to proceed OR provide a way to repeat dynamically presented information.**

Dynamic Displays.

Unlike a page of a text book a computer screen is a dynamic medium. We have the added dimension of time which allows text to be displayed or erased at will. Furthermore text can be black-on-white or white-on-black (inverse). Text can flash. Text or graphics can be animated and be moved about the screen to show relationships. Too many educational programs assume these features are only for arcade games but do not take advantage of them for tutorial programs. These dynamic characteristics, when used in an instructionally relevant way, can do much to increase the effectiveness of the instruction.

Timing of text output can be used for stress or to improve readability. In frames A and B of figure 7 the stress patterns in poetry and prose are indicated by displaying a passage syllable by syllable where each syllable is accompanied by a high sound when stressed or a low sound when unstressed. Frame D used timing to emphasize the attributes of the various kinds of meter. Rather than putting the entire diagram up at once the attributes are put up one by one with a pause between so that the student will focus his or her attention on the attributes one by one. The names of the kinds of meter are displayed one by one so that the student will focus attention on each name as it appears. This timing of text output makes the presentation more easily understood because the student's reading pattern is directed rather than left to chance.

Flashing text can also be used to direct attention and prepare the student for that which is to come. Look at figure 7, frames C and D. In these frames examples of the different types of meter are presented to the student one after another. If the words appeared without the direction to listen the student would not be prepared and would likely miss the use of sound to show the stress pattern. In this display the word, "LISTEN", and the arrow pointing to the box where the words will appear, flash for a second prior to the word appearing. In this way the student is anticipating the sound and the word when it appears.

If there is not an instructional purpose the use of these dynamic characteristics can be very distracting. Text which flashes or animation which occurs with no apparent purpose distracts rather than enhances learning. Look at figure 1. The cute cartoon characters which are used to indicate a correct answer soon lose their value. On the other hand the animation which occurs in figure 4 enhances the idea which is being taught by directing the students attention to the critical aspects of the display.

Computer displays, unlike textbooks, can be dynamic. Timing can be used to direct reading behavior, direct attention to particular portions of the screen or to stress certain ideas. In short, **use dynamic displays in which timing of text output, inverse text, flashing and animation are used for stress and emphasis.**

Display Planning.

When text books are put together a graphic designer often lays out the pages for aesthetic appearance and consistency. Effective layout can greatly facilitate the learning which can occur from the printed page. Too often, with computer displays, the need for graphic design is overlooked because the production process is simplified and steps which are usual in the production of books are eliminated. As a result too many programs look like rough drafts of printed material rather than carefully planned displays.

The screen should be planned as a unit. Each element, whether letter, word or graphic, should be carefully placed on the screen at the location that the designer selects. Furthermore the designer should have a purpose for placing information on the screen. If this purpose is aesthetic more pleasing appearing displays will result. If this purpose is instructional as well as aesthetic more learning will result. Following are a few characteristics of good screen design which should be considered. There is space here to identify only a few of the elements of good screen design.

Dark on light. The nature of a cathode ray tube makes turning on dots the natural way to write to the screen. Hence white (or lighted) letters on a dark screen became the norm. After one has worked at a cathode ray tube for many hours lighted letters and a dark screen began to appear as normal. However, for a student who may be learning from a computer for the first time light letters on a dark screen seems very unnatural. They are accustomed to seeing dark letters on a white page. Consequently it may be more desirable to paint the screen light and cause the print to appear as dark letters on a light screen.

Look at figures 1 and 2 which use light letters on a dark screen. Does this seem confined or closed in? Now look at figure 6 which uses dark letters on a light screen. This gives an open less restrained feeling. Which appears more natural? It should be noted that for clarity on the printed page some of the NOT-SO-GOOD poet program (figures 2 and 3) were printed in inverse. That is, the program used light letters and figures on a black background. We feel these programs would be easier to read if the programs used the black-on-white appearance of the printed page as they appear here. In short, **dark letters on a light screen will appear less confined and more natural to the student.** It is interesting to note that the new APPLE Macintosh uses dark letters on a light screen.

White space. Printed media has cost. A book of 100 pages costs more for materials than a book of 50 pages. To conserve costs (not to mention ease of use) printed materials try to put more on the printed page. In fact, because of some unscrupulous home study businesses, the postal service has standards about how much print must appear on a single page of instructional material sent through the mail. It is unlawful to send materials through the mail that does not meet this minimum. You cannot make the materials look to be more by putting fewer words on a page. A computer display on the other hand is free, that is, 20 displays cost no more than 1 display in terms of paper. Hence there is no good reason to fill the screen with text. Conversely there are often good instructional reasons to display only what the student needs at a given time on the screen. After a given piece of text or graphic has served its purpose it should be removed. Leaving it on the screen merely clutters the display with unnecessary and possibly distracting information.

Look at figure 2. Each of the screens are busy and full of text. Rather than inviting the student to read them they seem to promote the feeling, "Do I really have to read all that text?" Furthermore, after the student reads the first paragraph it (or worse, part of it) remains on the screen. This is unnecessary information which merely fills the screen and makes it more difficult to find the new paragraph.

Contrast the displays in figure 2 with the displays in figure 6. In figure 6 the displays were carefully planned and sentences appear only where the designer intended for them to appear. There is plenty of white space between sentences making them easier to read. In contract, the displays in figure 2 were determined by the built-in computer functions of word-wrap at the end of the lines and scrolling. The author merely entered the information and let the computer determine the display. In short, **leave plenty of white space and erase information when it is no longer needed.**

Short lines and natural phrases. In the early days, before 80 columns were common place, I have often heard designers of educational software complain that there was not enough space on the screen. Only 24 lines of 40 characters seems very confining when we are used to up to 60 lines of from 70 to 80 characters. However, the discipline of stating things more concisely may actually improve the instruction if each of the ideas to be presented are carefully considered and the words carefully chosen. When, as a designer, you realize that it is not necessary to start every line at the left margin and fill to the right margin then you begin to realize that the placement of text on the screen can actually enhance comprehension. Since we are no longer constrained to use the page economically we are free to use the layout to convey the message that we are trying to present.

Look at figure 7, frame B. Notice the response supplied by the computer to the rhetorical question, "What is a stressed syllable?" The stem of the statement is on the second line, "A stressed syllable is". Then each of the important ideas are expressed one per line, "louder", "emphasized", "higher in pitch". This listing tends to emphasize the individual characteristics so that they are not lost in a long string of prose. A similar use of short lines with a single natural phrase on each line is illustrated in the last part of this same frame. The stem of the statement is on the first line, "Stressed syllables are". The two ideas related to stressed syllables are emphasized by placing each on a separate line and indented from the stem, i.e., "higher in pitch" and "often found by reading aloud". A similar use of short lines with one natural phrase or idea per line is illustrated in frame C of this same figure. The stem appears on the first line, "A POETIC FOOT is". The characteristics are separated with one idea to a line and indented considerably, "a group of 2 or 3 syllables", "where 1 syllable is stressed", " and 1 or 2 are not stressed."

The poetic meter program sometimes does not use complete sentences when the idea can be presented with a list of abbreviated phrases. Some educators feel that educational software should demonstrate "good language use" and should therefore avoid sentence fragments and other incorrect usage. The use of short lines where one phrase appears per line can still be used as illustrated by frame. In short, **use short lines and separate natural phrases or ideas on each line.**

Fill justify. The thing that impressed me most when I first started to use a word processor was the fact that by merely giving a simple instruction I could have my document printed with right hand justification as well as left hand justification. The resulting document looked so professional, almost like it had been printed by a press. It has since come to my attention that some research has shown that a ragged right edge is easier to read. In fact the aesthetics of straight margins on both sides of the page may be detrimental to learning. What looks nice when a page is opened may in fact make it harder for the message to come across. If this is true for the printed page it is even more true for a computer screen. In short, **don't fill justify text on the screen.**

Upper case. In the early days of personal computers machines appeared which had no lower case alphabet. Many programs developed for these early machines were done in all upper case letters. Unfortunately, the BASIC programming language that still exists on many machines will allow lower case letters only with some difficulty. As a result it is sometimes easier to program in all upper case. However, the data shows that children often use the shape of the word as a cue to its meaning. When words appear as all upper case this cue is gone and reading becomes more difficult. There is no real excuse for professional programs to use all upper case and there is considerable evidence that all upper case presentations are more difficult to read and comprehend.

An isolated word or title in all upper case tends to make it stand out and is one form of creating emphasis. The use of all capitol letters for emphasis probably enhances the instructional impact and is desirable. But if all the letters are capitols this emphasis is lost. In short, **do not present information in all upper case except for emphasis.**

Text style variety. The printed page is often made more pleasing by the inclusion of different styles of text. If this use of text style is haphazard and not related to the instructional purposes of the presentation it is probably distracting. On the other hand use of a different style of text can enhance the presentation if it is used for emphasis or to indicate that the text appearing in the new style has a different purpose. Newer personal computers allow different text styles on the screen even for ordinary document creation.

The examples included with this paper have not made wide use of different text styles except the use of inverse (technically reverse inverse). In figure 7, frame A the passages are shown as white-on-black to set it apart from the instructional text which is black-on-white. This is consistent throughout the program. All of the sample words and poetry is shown inside a black box as white-on-black. This helps the student to find the illustrative material and separate it from the instructional text. In short, **use a variety of text styles to indicate a different kinds of messages.**

HUMAN FACTORS

Human factors refer to those characteristics of the software that make it easy for the student to use. The difficulty of using many computer programs has become recent folklore. It is frequently necessary to memorize a long list of control characters or other commands in order to interact with the program. The most recent advances have attempted to deliver software that requires little or no knowledge of computers; software with which the user can interact by pointing. In this section we have suggested some of the human factor concerns which will make educational software easier to use and thus result in more efficient learning.

Control of location.

Many educational programs have the characteristic of an adventure game. The student has wondered into a maze and cannot see the way in or the way out. The only choice available is to keep working in the hope that the light at the end of he tunnel will soon appear. The problem with this approach is that the student may find that he or she already knows what is being presented and would like to skip ahead to the next section. Or, worse, the student is having difficulty and would like to return to a previous section. Many students have learned that a quick survey through the material before beginning more serious study greatly facilitates the ability to integrate the new information with that which is already known. When there is no indicators at to where you are or where you are going this survey activity is impossible.

By the use of indexes and tables of contents books enable us to preview, survey, find an isolated piece of information, skip ahead or review. If we do not provide similar devices in educational software we have created not more flexible but less flexible educational materials. Some possible devices for allowing more student control over the materials are as follows:

Table of Contents. Like a book a table of contents is a useful tool for an educational program. This should list the major topics to be covered and provide a way for the student to "turn to" a given topic. Look at figure 5, frame B. In the Poetic Meter program after the title page the student is presented with a list of the sections of the program. In his program the student is required to keep track of their own progress (there is no student manager program) and this table of contents was necessary to enable the student to pick up where he or she left off. But even if this was not necessary having the table of contents as an option would facilitate preview and review. In short, **provide a way for the student to skip to the major sections of the program in order to preview, review or repeat portions of the material.**

Location indicators. A table of contents provides a map of where to go and the mechanism for getting there but do you know where you are when you get there. The size of a software program is transparent to the student. It is hard to turn to the last page and see how much more you have to study before it is convenient to take a break. Books use page numbers. On the table of contents page you can observe that a given section is 10 pages long. Then by monitoring the page numbers you can tell how much of the section is left. Some similar concept is needed for educational software. Perhaps a page number is sufficient.

Look at figures 6 through 11. Notice that each display has a number at the bottom of the screen. This is an attempt to tell the student where he or she is in the program. By comparing this number with the table of contents the student can determine where they are in the total program. In short, **provide some sort of location indicator so that the student knows where he or she is in the total program.**

Pages. When studying a book if a student gets distracted it is easy to turn back to the last page and reread the material to pick up on the train of thought. Too often in educational software after the return key has been pressed the last presented material is gone never to be retrieved unless the program is started over. While a table of contents allows preview or review of whole sections of material there is need for a mechanism for short term review or preview. The student must be able to "turn back" to the previous page, to "turn forward" to the next page or, in the case of dynamic displays repeat the current page.

Look at figure 6, frame D and figure 7, frame B. At the conclusion of each page in the Poetic Meter program the student is given the option of going back to the last page by pressing the left arrow key, going on to the next page by pressing the right arrow key or repeating the current page by pressing the return key. In short, **allow the student to**

"turn" the pages by going back to the last page, repeating the current page or going forward to the next page.

Keyboard anxiety.

More people are afraid of keyboards than are afraid of computers. Typing has been seen as a skill for typists (a job description rather than merely a qualification), clerks and secretaries. It is still taught as an elective course in high school (usually in the business department). It is apparent that like handwriting typing will be a basic skill in the future but for the time being we must realize that the majority of the population do not have keyboard skills. This means that lengthy keyboard input is likely to cause anxiety and frustration. Therefore programs should be designed that minimize the amount of information required from the keyboard.

The latest technology in computer programs uses pointing as the primary input device. Pointing, using a mouse or an easily located key, is much easier than finding a particular letter or number on the keyboard. A mouse is a hand held device which rolls on a desk moving a pointer on the screen. A button on the mouse selects the function which the arrow points to on the screen. Pointing is much easier than typing a word or phrase. There are times when we want unstructured input from the student. Figure 6 represents this type of situation. Other times when we are involved in selecting a menu or choosing a response to a question pointing will facilitate the student's interaction. Figure 5 illustrates pointing on a menu. The student merely presses the arrow keys to move the pointer up and down the menu. When the arrow points at the choice that the student wishes to select he or she merely presses the return key. A similar pointing procedure is used in selecting examples and answering practice problems. Figure 8, frame A, shows the use of a pointing device for selecting the type of meter the student wishes to see illustrated. Pressing the space bar moves the pointer (the brackets around the choice in this case) which rotates past each of the choices in turn. The return key gets the choice. Figure 9, frame A shows a similar pointer for practice. In this case the student uses the space bar to select the type of meter illustrated. Pressing the return key accepts the student answer.

Many students get frustrated when required to use the keyboard and this anxiety interferes with learning. In short, **minimize unnecessary typing by using a pointing device whenever possible.**

Advisor.

One aspect of effective teaching is to monitor the student's process and progress and to provide advice for the student to guide further study efforts. This type of monitoring function is difficult to implement in the class room where there are usually many children for every teacher. One advantage of the computer is that it can not only present information to the student but it can gather information from the student. This student generated information can then be used to monitor the student's progress and can serve to trigger advice to the student which may facilitate his or her learning.

The most obvious thing to monitor is the number of practice problems that a student misses. However, there is much other information which can be used to determine if the student is using the instructional program in an effective way. Some of these include monitoring the number of examples studied by the student, checking to see if the student has used the resources provided by the program as well as checking on practice performance.

Look at figure 8, frame B. This display is shown to the student if he or she presses the escape key to terminate the example portion of the program before we as the designer feel that an adequate number of examples have been studied. In fact there is a prior message in this section of the program. If the student tries to escape from the example section before looking at a minimum of two examples of each type a message says, "You must study at least 2 examples of each type." In this example display the amount of prompting material is slowing decreased. Early examples are shown divided into syllables and the word or phrase is chanted for the student. After a few examples the colon which divides the words into syllables is removed and the words are no longer divided into syllables for the student. After a few more examples the chant is turned off and the student merely sees a word or phrase of the type indicated without additional prompts. If the student escapes before seeing at least 2 nonprompted examples of each type the warning message of figure 8, frame B appears. If the student does exceed this criteria then the danger frame is replaced by a frame which say, "GO, you should be ready for practice." Notice that in either the case of the danger or go advice the student has the option to follow the advice or ignore the advice. The black stripe at the bottom of the page gives the student the option of returning for more examples by pressing the left arrow key or going on to practice by pressing the right arrow key. If the student elects to return to the example section of the program the software remembers where the student left off and begins presenting examples from this point.

A similar advisor function follows practice. Look at figure 9, frame C. In this case the student is given a score board showing how many of each type of meter were classified incorrectly. If the student misses more than 2 of any type the warning includes a stop sign and the advice, "You should study more examples." If the student misses less than 1 of each type the display contains a "GO" sign and the advice, "Ok to study the next page". If the performance is between these two criteria levels the display contains a "CAUTION" sign and the advice, "Perhaps you should study more examples." The student is given three options. Turn to the next page by pressing the right arrow. Go back to the example section by pressing the left arrow. Do some more practice by pressing the return key. Again the student has the option to override the advice.

Computers provide the mechanism to monitor not only the student's performance but all of the student's learning activities. The teaching function of advice can easily be implemented. In short, **monitor the student's activity and provide advice when potentially decremental action is taken. In most cases provide a mechanism for the student to override the advice.**

Learner Control.

In the early days of computer assisted instruction advocates often stated that someday we would have a computer system which could assess the readiness and aptitude of the student and then present that information which was an optimal match to the individual student. Even if it were possible there remains some serious ramifications of this assumption. If a significant part of the curriculum was constructed around such adaptive instruction the gains on the part of the student in the subject matter being taught might be considerable but the side effects could be catastrophic. The students in such a system would be spoon fed and might never learn to feed themselves. The real world is seldom so accommodating. If one of the primary aims of education is to assist the student to be an independent learner and acquire additional learning by self teaching then our highly adaptive CAI system may turn out to be maladaptive after all. Students who have a major part of their learning on such an accommodating system would not learn the important metaskills necessary to learn from less than optimal information.

A better approach might be to teach the student to adapt the system to themselves rather than for the system to adapt itself to the student. The mechanism for this student adaptation is to provide learner control so that many decisions about the next best display is left to the student. Rather than big brother providing the information for the student the student must select that information that he or she thinks is best for himself or herself.

Text output. In an earlier section we discussed student control of text output. This is an obvious area where students can determine for themselves when they are ready for the next display. Exceptions might be in timed tests or when speed reading is the objective but usually the student should indicate, "I'm ready to continue."

Number of examples. Research has shown that most students can determine how many examples they need to study in order to understand an idea. The number of examples required by different individuals may differ considerably but if an individual is shown less than he or she feels is needed there will be a decrement in performance. On the other hand, showing them more than they think is needed serves only to prolong the instruction without any accompanying increase in performance.

Look at figure 8, frame A. In this presentation the student can determine the number of examples needed to understand each type of meter. When the student feels that enough have been studies he or she presses the escape key and jumps to the practice section. Note that the advisor function previously discussed comes into play to monitor the student's use of learner control. If in our opinion the student might be using learner control to avoid what might be necessary study the program provides a caution to the student. However, the student is still in control and can usually override the advice. In short, **allow students to select how many examples they need to study.**

Optional help. Students differ significantly in the amount of elaboration they need to understand an idea. Some students need a step by step explanation while others understand after the general idea has been presented or after they have seen an example. It is inefficient and may cause frustration for the student if forced to work through every step of a detailed analysis. On the other hand if this detailed analysis is not available some students will be lost.

Look at figure 10. This sequence of frames illustrates the detailed analysis of the poetry scanning procedure. First, the student is advised to read the poem aloud and take a guess a the type of meter involved (frame A). The scanning editor then "chants" the poem for the student using the sound to indicate the stress pattern involved. In frame B the student is advised to separate multisyllable words into separate syllables using the colon. In frame C the student is advised to mark the stressed syllables using the underline. And in frame D the student is advised to divide the passage into poetic feet using the up arrow. Finally the student is asked to observe the pattern of stress in each foot and indicate the type of meter involved.

Some students need to repeat this detailed analysis many times while others will get the idea very quickly. Students can jump to the practice at any time by pressing the escape key. In the practice section (see figure 11) the student is asked to classify the type of meter involved in the passage (frame A). In frame B the word RIGHT or WRONG is flashed in the upper right part of the screen to indicate whether or not the classification was correct. The passage is then chanted for the student to allow comparison. The correct choice is not indicated to the student when he or she is wrong. The student can then choose whether or not to engage in the more detailed analysis. Pressing the [?] key causes frame D to appear and then follows a detailed scanning analysis such as was presented in the example section.

Because students differ in the amount of detailed elaboration they need they should be given a choice as to whether or not they wish more explanation. Forcing all students through a linear path consisting of the same level of detail may be boring for some students because the pace is too slow and frustrating for others because the pace is too fast. In short, **provide optional help, don't force every student through the most detailed presentation.**

Escape. Sometimes students are interrupted while studying. There are many causes for such interruptions. If a student is in a long sequence they are very hesitant to merely pull out the disk and turn off the computer. This may mean that they must start over the next time or they may fear that this will somehow ruin the program. The program should provide a way for the student to return to the menu or some other starting place.

Poetic Meter allows the student to escape from any sequence by pressing the escape key. Usually the consequence of such an action is indicated by the program. In the example sections the program uses the escape as a means of learner control for the student to tell the program that they have seen enough examples and wish to go on. The advisor warns the student if we feel that the student should study more examples. This program does not have a student manager so if the student uses escape to leave the program the program starts over when the program is restarted. Including a student manager is better. This provides a way for the program to restart the student where they left off without the student having to keep track and use the menu to find their place.

Everyone suffers from interruptions and needs a way to exit in an orderly way short of pulling the plug. In short, **provide a means of escape from any lengthy activity but advise the student about the consequence of such an escape.**

Directions.

Those of us who have been in the computer business for some years have come to accept a number of conventions that seem like second nature. For example, everyone that has used a computer very long soon learns that to enter a response it is usually necessary to press the return or enter key. This is so obvious to us that we usually feel that to inform the student about this convention is unnecessary. However, more and more first time users are beginning to use the computer. What may be obvious to an experienced user is not at all obvious to the first time user. The learner will have a more pleasant experience if the program contains complete directions about what to do and when.

Often documentation will contain the necessary directions but if they are at all complex, as in editors where there are several options available, most users find that it is necessary to have a summary card in order to remember which commands go with the program. If possible the current options should be available to the student as options on the screen.

Look at figure 9, frame A. This is a simple exercise but the directions for the student are shown on the screen so that what is necessary to make the selection is clearly indicated. The Poetic Meter program makes a poor assumption on figure 6, frames A and B. It assumes that the student will know that when the cursor appears he or she should enter an answer and then press the return key. It would be an improvement if this direction appeared on the screen at least for the first few times the student is expected to make some constructed response to a request from the program.

Don't create interfering directions. Sometimes in an attempt to make it very clear to the beginning user the procedures become very inconvenient for the experienced user. This is probably less of a problem in an educational program than it is in some utility such as a word processor or data base. Nevertheless we should be careful to balance directions so that the program is easy to use for the beginner but convenient to use for the more experienced user.

Make the directions as natural as possible. There is nothing more difficult than a long menu that is numbered. The user must look at the menu, get an arbitrary number, and hunt for the number on the keyboard in order to make a selection. Mnemonic choices using the first letter of the word may be easier to remember. Pointing, as already mentioned, is easiest of all. Keys should be used in a natural way or in ways consistent with other systems. Unique combinations of key presses make it unnecessarily difficult for the student. Even the advise to the student should use already known symbols. In the Poetic Meter program the advisor uses "STOP", "CAUTION" and "GO" as ways to get the student's attention quickly and to leave little ambiguity about what is being said. (See figure 8, frame B and figure 9, frame C.) In short, **provide adequate directions including all of the options available to the student. If possible make these available on the screen or accessible with the press of the [?] key. Use the most natural procedure.**

Program Structure.

This paper has not dealt with programming design but it is necessary to border on the architecture of the program for one final human factors concern. While the power and capacity of personal computers seems to increase daily with each new product on the market, there are still severe limitations of space and size. These limitations can interfere with effective educational design. Some popular programs are very annoying because they continually access the disk drive. For the student it seems like half the time is spent waiting for the computer to load the next part of the program. The polite message, "Thank you for waiting." gets very old in a hurry. By carefully structuring the program these long delays in processing or accessing information can be minimized. It is especially important that there be no delays between the time when a student enters information and the computer reacts to that information. If there is a delay the student often feels that the computer is not operating properly and may try to enter additional information. Whenever possible the next part of the program should be anticipated and loaded during a time when the student is busy with other activities on the screen. Thus when the student indicates that he or she is ready to continue the next part of the program is already in memory and easily displayed to the student. In short, **plan disk access to avoid long waits while the computer retrieves information.**

In Conclusion.

Education Secretary Terrel Bell has recently criticized educational software as "electronic page turning [that] hasn't been designed to do a good job of interacting with the mind of the student." He further states that computers could be used as "slave mechanisms" to check spelling, punctuation and sentence structure but he doubts that computers could ever teach writing. As a designer of educational software (including software that teaches writing and poetic meter) I was ready to go to battle with the good Secretary. Then I realized that a random sampling of educational software might very well create the impression that computers are pretty limited in what they can teach. It is time that the consumers of educational programs become more discerning and refuse to purchase software that does not meet at least the minimum standards of acceptable instruction. This paper has attempted to provide a few suggestions about some of those minimum standards.

SUMMARY

Instructional Strategies

The first section of this paper identifies design strategies which are believed to have a direct impact on the amount of learning that occurs as a result of instruction. Instructional design strategy involves the arrangement of content elements and other information in such a way that learning is facilitated. The following guidelines are suggested:

1. Avoid merely putting text on the screen; avoid mere page turning.

2. Avoid generality-rich but example-poor presentations. Each idea should be presented by a generality, examples and practice.

3. Avoid remember only practice.

4. Use attention focusing devices to relate examples to generalities and to point out critical characteristics of the illustrative material.

5. Promote active mental processing by asking rhetorical questions and engaging the student in a conversation which requires constructed (as opposed to multiple choice) responses to which we do not provide right-wrong feedback but rather an anticipation of a reasonable reaction.

6. Provide expository examples as well as practice.

Display Techniques

The second section of this paper identifies screen display techniques which are believed to facilitate the students ability to interact with the materials. Screen display refers to the way that information is exhibited on the monitor. The following guidelines are suggested:

1. No scrolling for educational programs.

2. The student should control text output. Never erase critical information until the student indicates readiness to proceed OR provide a way for the student to repeat dynamically presented information.

3. Use dynamic displays in which timing of text output, inverse text, flashing and animation are used for stress and emphasis.

4. Dark letters on a light screen will appear less confined and more natural to the student.

5. Leave plenty of white space and erase information when it is no longer needed.

6. Use short lines and separate natural phrases or ideas on each line.

7. Don't fill justify text on the screen.

8. Do not present information in all upper case except for emphasis.

9. Use a variety of text styles to indicate different kinds of messages.

Human Factors

The third section of this paper identifies some human factors characteristics which will make educational software easier to use and thus result in more efficient learning. The following guidelines are suggested:

1. Provide a way for the student to skip to the major sections of the program in order to preview, review or repeat portions of the material.

2. Provide some sort of location indicator so that the student knows where he or she is in the total program.

3. Allow the student to "turn" the pages by going back to the last page, repeating the current page or going forward to the next page.

4. Minimize unnecessary typing by using a pointing device whenever possible.

5. Monitor the student's activity and provide advice when potentially decremental action is taken. In most cases provide a mechanism for the student to override the advice.

6. Allow students to select how many examples they need to study.

7. Provide optional help, don't force every student through the most detailed presentation.

8. Provide a means of escape from any lengthy activity but advise the student about the consequence of such an escape.

9. Provide adequate directions including all of the options available to the student. If possible list the available options on the screen or make them accessible with the press of the [?] key. Use the most natural procedure.

10. Plan disk access to avoid long waits while the computer retrieves information.

FIGURES

Figure 1 Drill and practice program.

Figure 2 NOT-SO-GOOD Poetic Meter.

Figure 3 NOT-SO-GOOD Poetic Meter continued.

Figure 4 EDU-WARE FRACTIONS with animation.

Figure 5 Table of Contents - Poetic Meter.

Figure 6 Presentation page 1 - Poetic Meter.

Figure 7 Presentation selected pages - Poetic Meter.

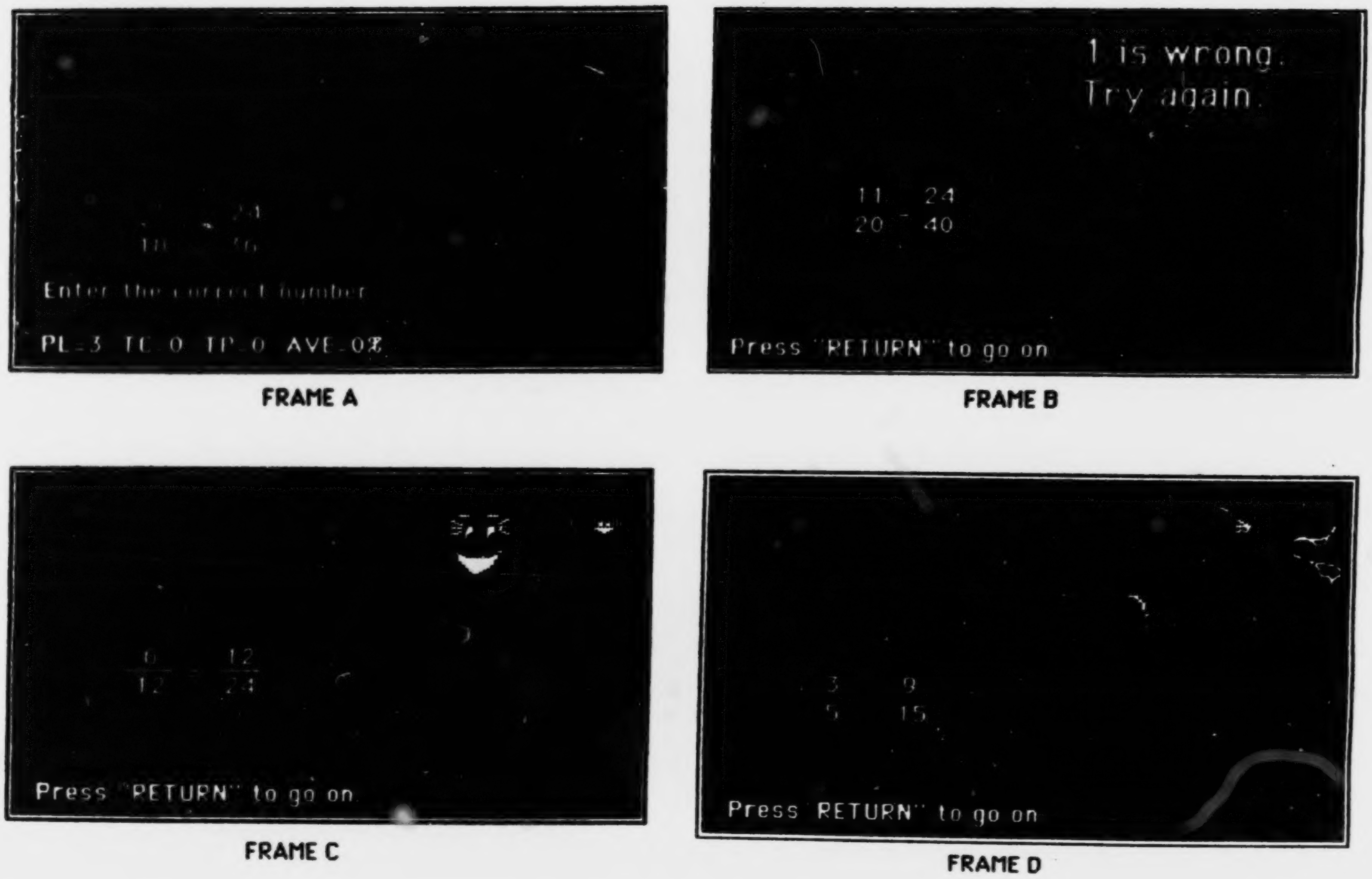
Figure 8 Word Example - Poetic Meter.

Figure 9 Word Practice - Poetic Meter.

Figure 10 Verse Example with scanning editor - Poetic Meter.

Figure 11 Verse Practice with scanning editor - Poetic Meter.

BEST COPY AVAILABLE



24
10 36

Enter the correct number

PL=3 TC=0 TP=0 AVE=0%

FRAME A

1 is wrong.
Try again.

11 24
20 40

Press "RETURN" to go on

FRAME B

6 12
12 24

Press "RETURN" to go on

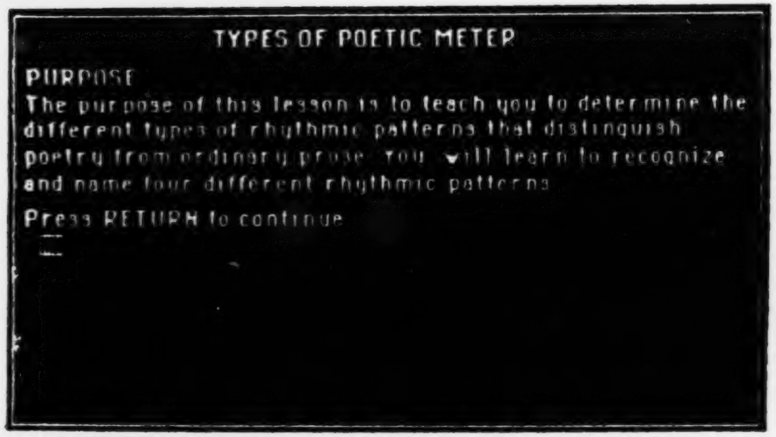
FRAME C

3 9
5 15

Press "RETURN" to go on

FRAME D

Figure 1 Drill and practice program.



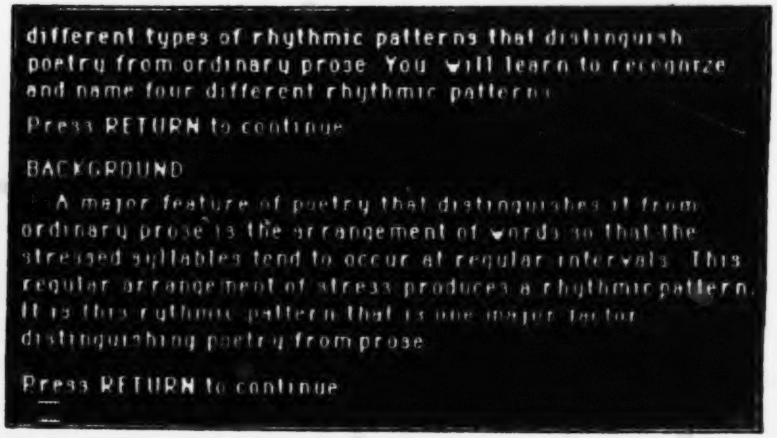
TYPES OF POETIC METER

PURPOSE

The purpose of this lesson is to teach you to determine the different types of rhythmic patterns that distinguish poetry from ordinary prose. You will learn to recognize and name four different rhythmic patterns.

Press RETURN to continue

FRAME A



different types of rhythmic patterns that distinguish poetry from ordinary prose. You will learn to recognize and name four different rhythmic patterns.

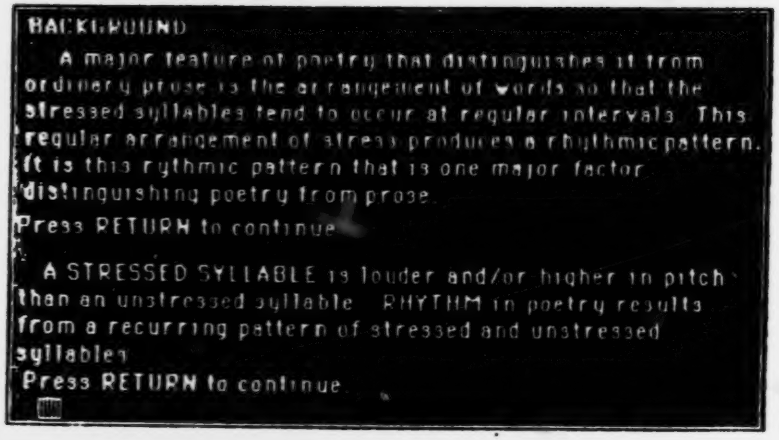
Press RETURN to continue

BACKGROUND

A major feature of poetry that distinguishes it from ordinary prose is the arrangement of words so that the stressed syllables tend to occur at regular intervals. This regular arrangement of stress produces a rhythmic pattern. It is this rythmic pattern that is one major factor distinguishing poetry from prose.

Press RETURN to continue

FRAME B



BACKGROUND

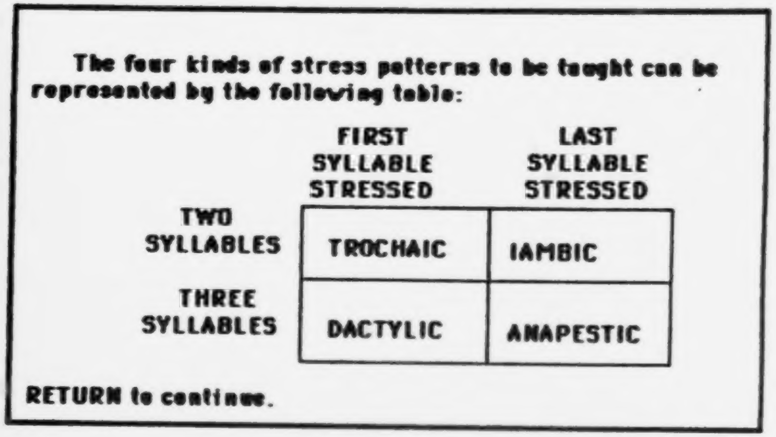
A major feature of poetry that distinguishes it from ordinary prose is the arrangement of words so that the stressed syllables tend to occur at regular intervals. This regular arrangement of stress produces a rhythmic pattern. It is this rythmic pattern that is one major factor distinguishing poetry from prose.

Press RETURN to continue

A STRESSED SYLLABLE is louder and/or higher in pitch than an unstressed syllable. RHYTHM in poetry results from a recurring pattern of stressed and unstressed syllables.

Press RETURN to continue.

FRAME C



The four kinds of stress patterns to be taught can be represented by the following table:

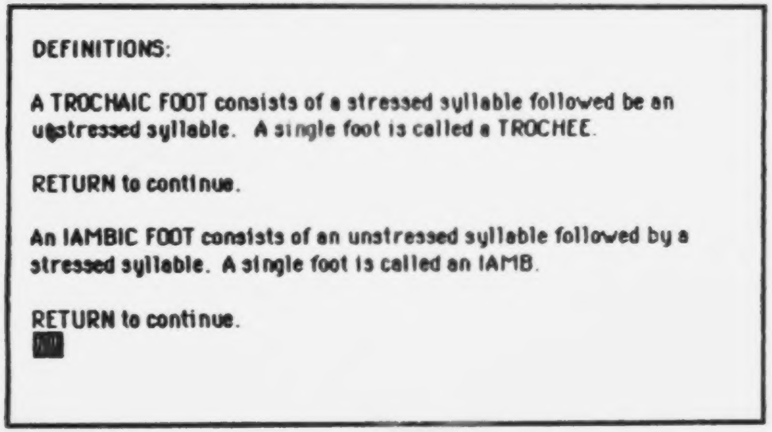
	FIRST SYLLABLE STRESSED	LAST SYLLABLE STRESSED
TWO SYLLABLES	TROCHAIC	IAMBIC
THREE SYLLABLES	DACTYLIC	ANAPESTIC

RETURN to continue.

FRAME D

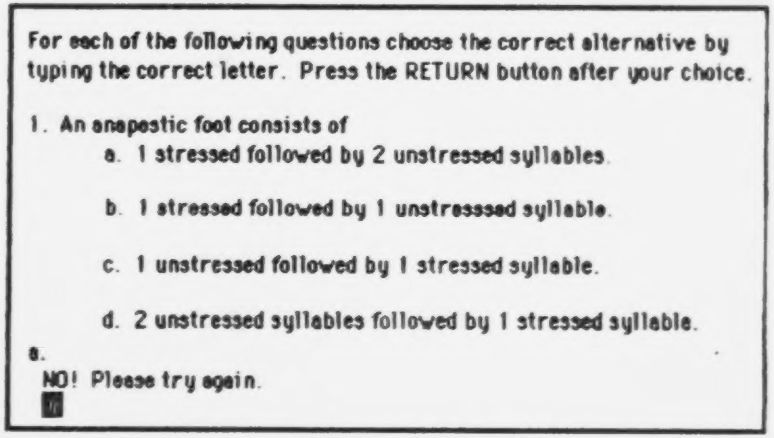
Figure 2 NOT-SO-GOOD Poetic Meter

BEST COPY AVAILABLE



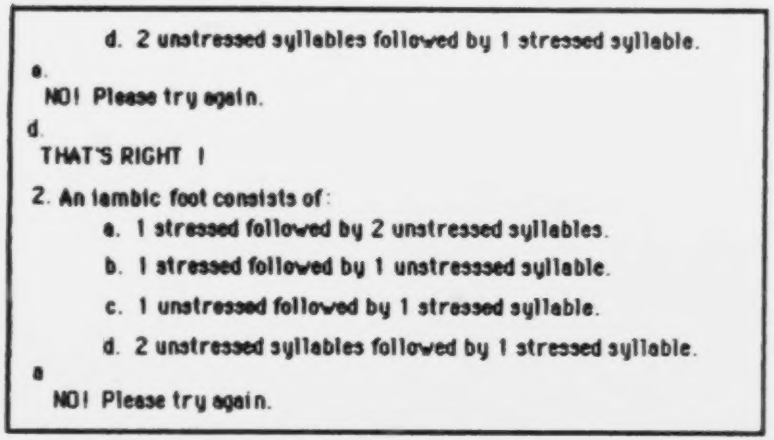
```
DEFINITIONS:

A TROCHAIC FOOT consists of a stressed syllable followed be an
unstressed syllable.  A single foot is called a TROCHEE.

RETURN to continue.

An IAMBIC FOOT consists of an unstressed syllable followed by a
stressed syllable.  A single foot is called an IAMB.

RETURN to continue.
```

FRAME A



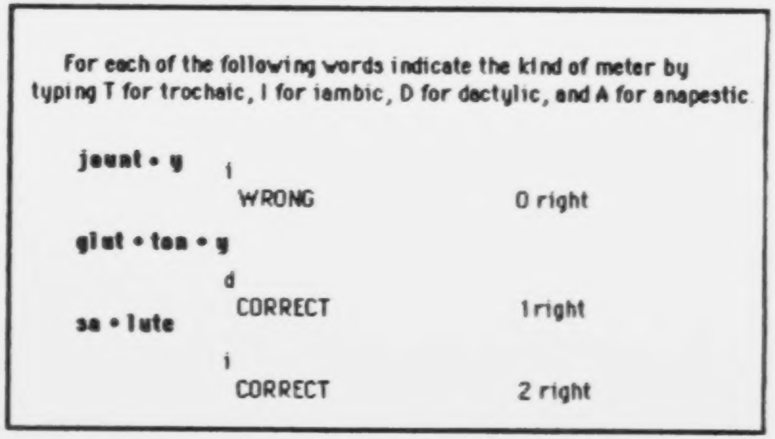
```
For each of the following questions choose the correct alternative by
typing the correct letter.  Press the RETURN button after your choice.

1.  An anapestic foot consists of
          a.  1 stressed followed by 2 unstressed syllables.

          b.  1 stressed followed by 1 unstresssed syllable.

          c.  1 unstressed followed by 1 stressed syllable.

          d.  2 unstressed syllables followed by 1 stressed syllable.
a.
 NO!  Please try again.
```

FRAME B



```
          d.  2 unstressed syllables followed by 1 stressed syllable.
a.
 NO!  Please try again.
d.
 THAT'S RIGHT  !
2. An iambic foot consists of:
          a.  1 stressed followed by 2 unstressed syllables.
          b.  1 stressed followed by 1 unstresssed syllable.
          c.  1 unstressed followed by 1 stressed syllable.
          d.  2 unstressed syllables followed by 1 stressed syllable.
a
  NO!  Please try again.
```

FRAME C

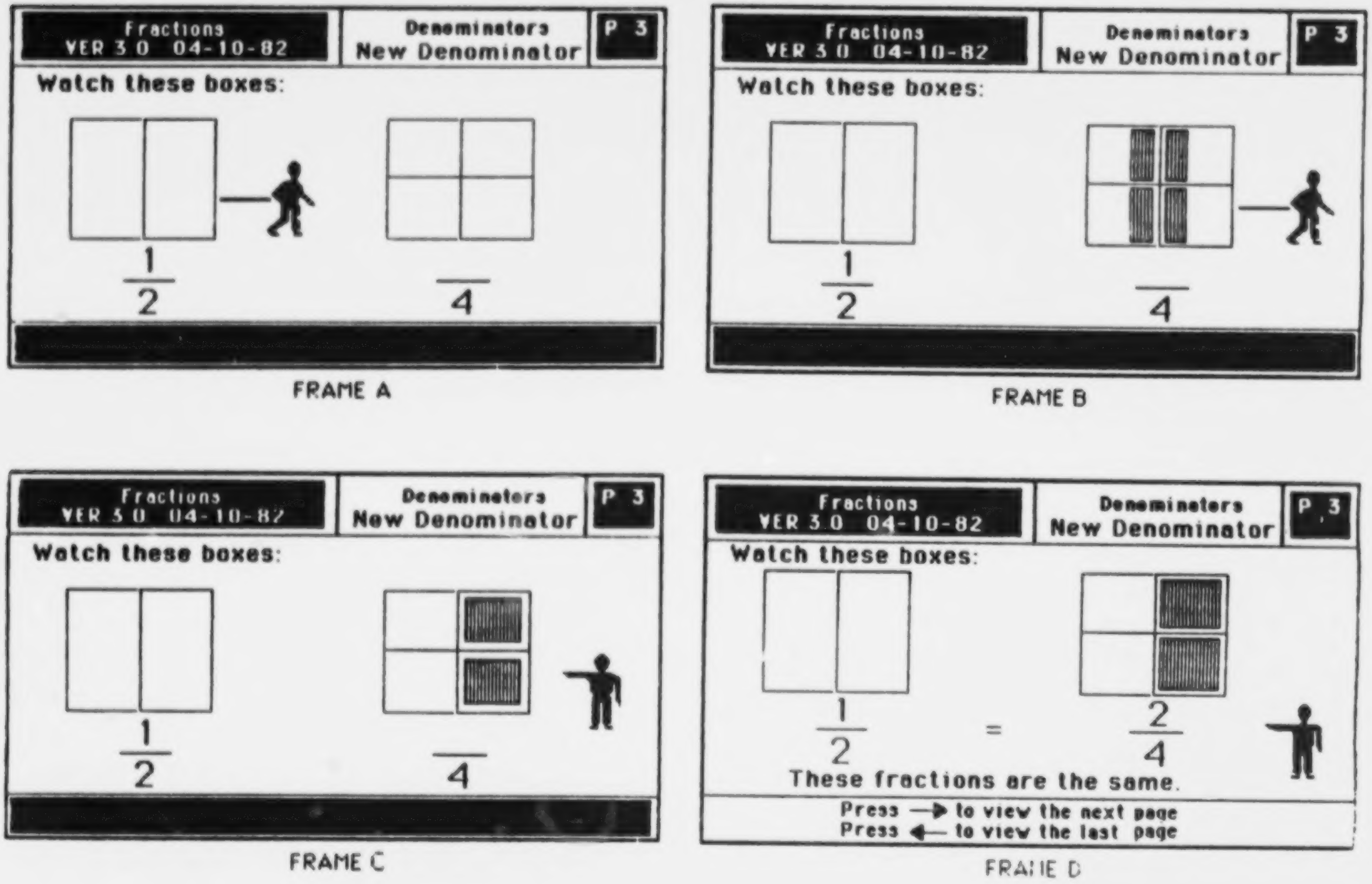


```
   For each of the following words indicate the kind of meter by
typing T for trochaic, I for iambic, D for dactylic, and A for anapestic.


    jaunt • y    i
                  WRONG                    0 right
    glut • ton • y
                 d
                  CORRECT                  1 right
    sa • lute
                 i
                  CORRECT                  2 right
```

FRAME D

Figure 3   NOT-SO-GOOD Poetic Meter continued

BEST COPY AVAILABLE



Fractions
VER 3.0 04-10-82
Denominators
New Denominator
P 3
Watch these boxes:
$\frac{1}{2}$
$\frac{\ }{4}$

FRAME A

Fractions
VER 3.0 04-10-82
Denominators
New Denominator
P 3
Watch these boxes:
$\frac{1}{2}$
$\frac{\ }{4}$

FRAME B

Fractions
VER 3.0 04-10-82
Denominators
New Denominator
P 3
Watch these boxes:
$\frac{1}{2}$
$\frac{\ }{4}$

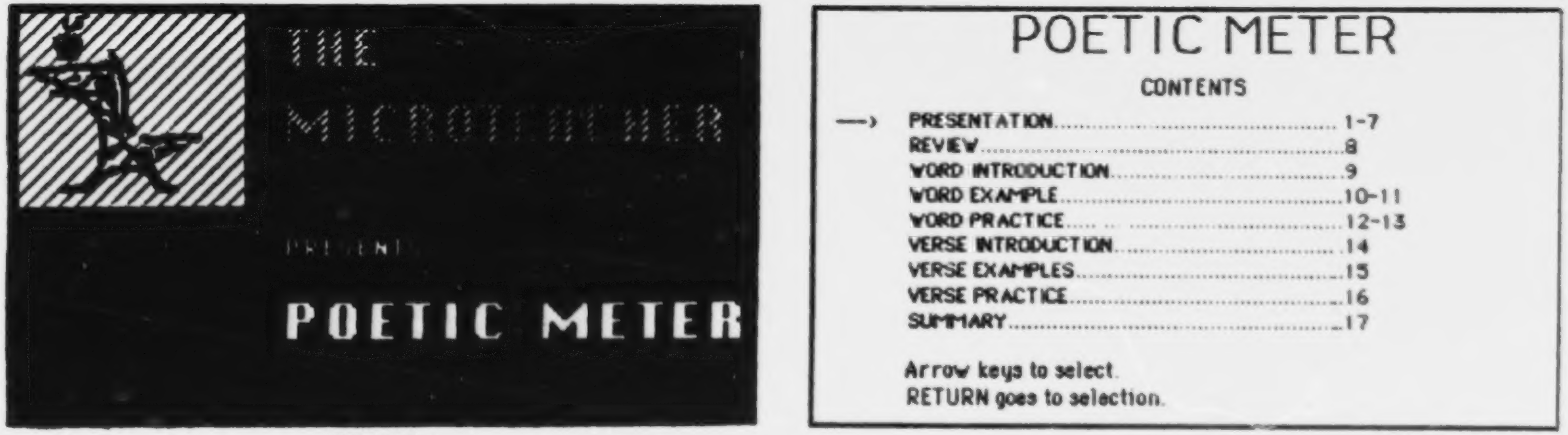
FRAME C

Fractions
VER 3.0 04-10-82
Denominators
New Denominator
P 3
Watch these boxes:
$\frac{1}{2} = \frac{2}{4}$
These fractions are the same.
Press → to view the next page
Press ← to view the last page

FRAME D

Figure 4 EDU-WARE FRACTIONS with animation

BEST COPY AVAILABLE



THE
MICROTEACHER
PRESENTS
POETIC METER

FRAME A

POETIC METER

CONTENTS

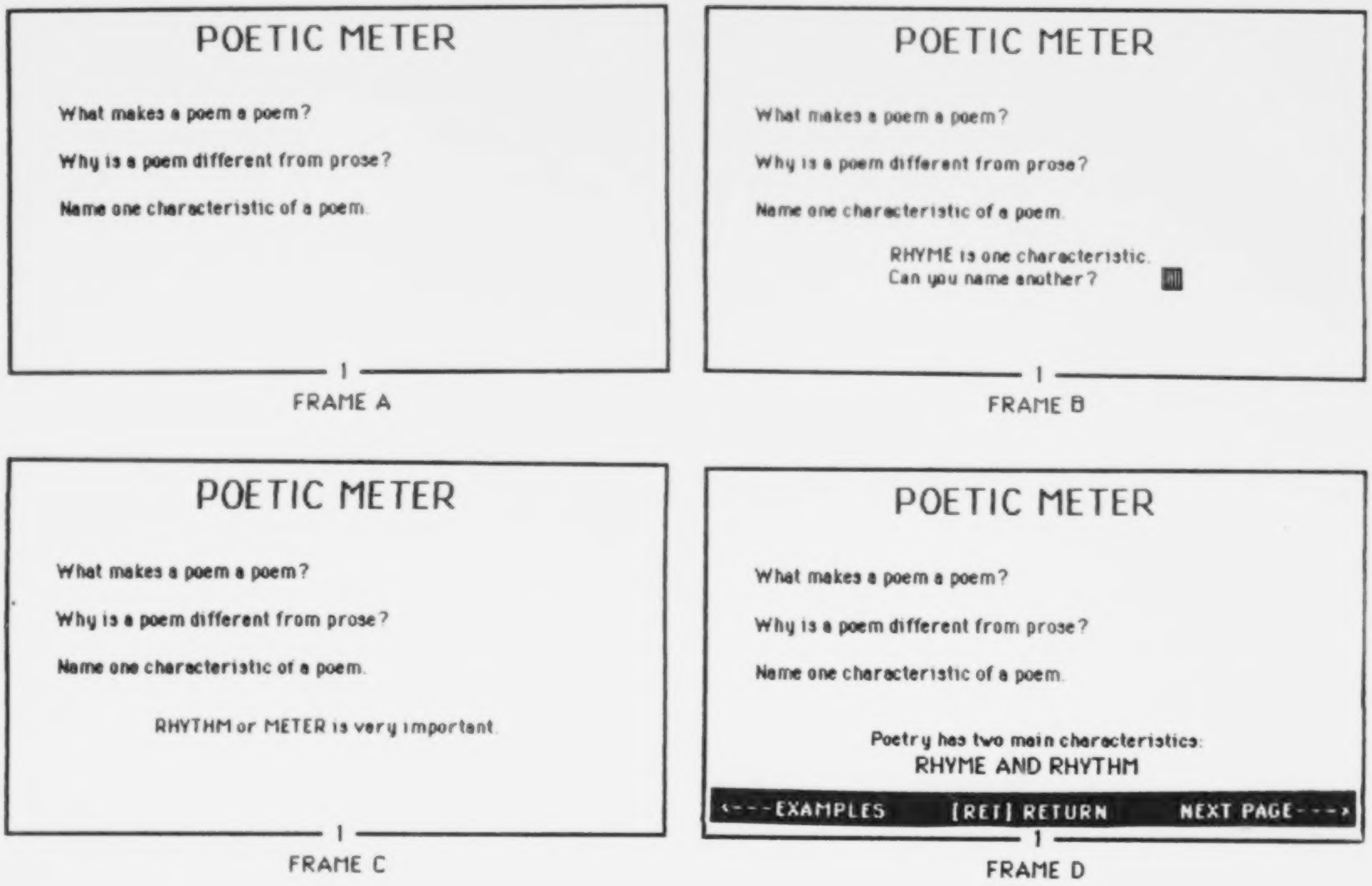
—› PRESENTATION 1-7
REVIEW 8
WORD INTRODUCTION 9
WORD EXAMPLE 10-11
WORD PRACTICE 12-13
VERSE INTRODUCTION 14
VERSE EXAMPLES 15
VERSE PRACTICE 16
SUMMARY 17

Arrow keys to select.
RETURN goes to selection.

FRAME B

Figure 5 Table of Contents - Poetic Meter

BEST COPY AVAILABLE



FRAME A

POETIC METER

What makes a poem a poem?

Why is a poem different from prose?

Name one characteristic of a poem.

1

FRAME B

POETIC METER

What makes a poem a poem?

Why is a poem different from prose?

Name one characteristic of a poem.

RHYME is one characteristic.
Can you name another?

1

FRAME C

POETIC METER

What makes a poem a poem?

Why is a poem different from prose?

Name one characteristic of a poem.

RHYTHM or METER is very important.

1

FRAME D

POETIC METER

What makes a poem a poem?

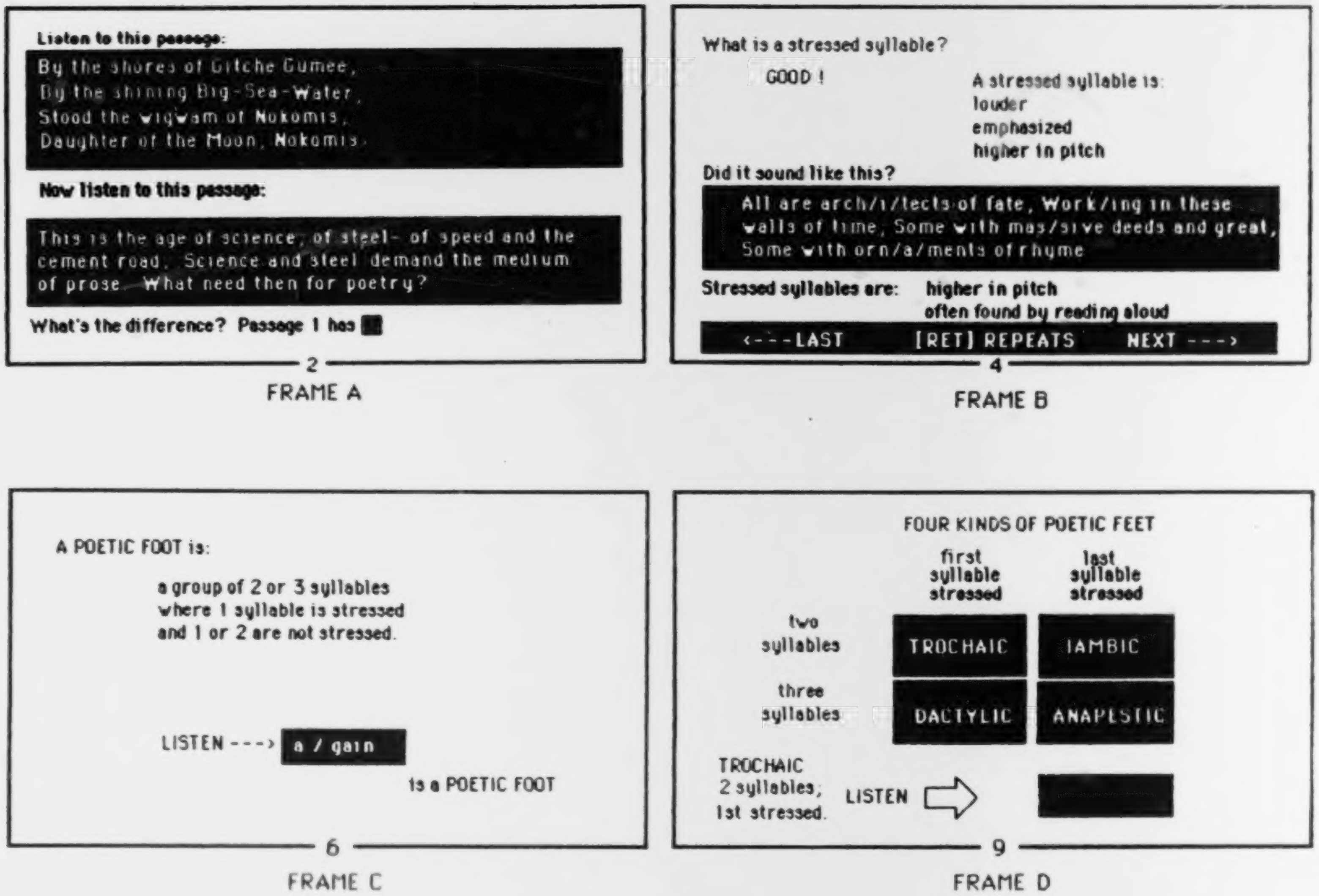
Why is a poem different from prose?

Name one characteristic of a poem.

Poetry has two main characteristics:
RHYME AND RHYTHM

1

Figure 6 Presentation page 1 - Poetic Meter



Listen to this passage:

By the shores of Gitche Gumee,
By the shining Big-Sea-Water,
Stood the wigwam of Nokomis,
Daughter of the Moon, Nokomis.

Now listen to this passage:

This is the age of science, of steel- of speed and the cement road. Science and steel demand the medium of prose. What need then for poetry?

What's the difference? Passage 1 has ▮

2

FRAME A

What is a stressed syllable?

GOOD !

A stressed syllable is:
louder
emphasized
higher in pitch

Did it sound like this?

All are arch/i/tects of fate, Work/ing in these
walls of time, Some with mas/sive deeds and great,
Some with orn/a/ments of rhyme

Stressed syllables are: higher in pitch
often found by reading aloud

<---LAST [RET] REPEATS NEXT --->

4

FRAME B

A POETIC FOOT is:

a group of 2 or 3 syllables
where 1 syllable is stressed
and 1 or 2 are not stressed.

LISTEN ---> a / gain

is a POETIC FOOT

6

FRAME C

FOUR KINDS OF POETIC FEET

	first syllable stressed	last syllable stressed
two syllables	TROCHAIC	IAMBIC
three syllables	DACTYLIC	ANAPESTIC

TROCHAIC
2 syllables;
1st stressed.

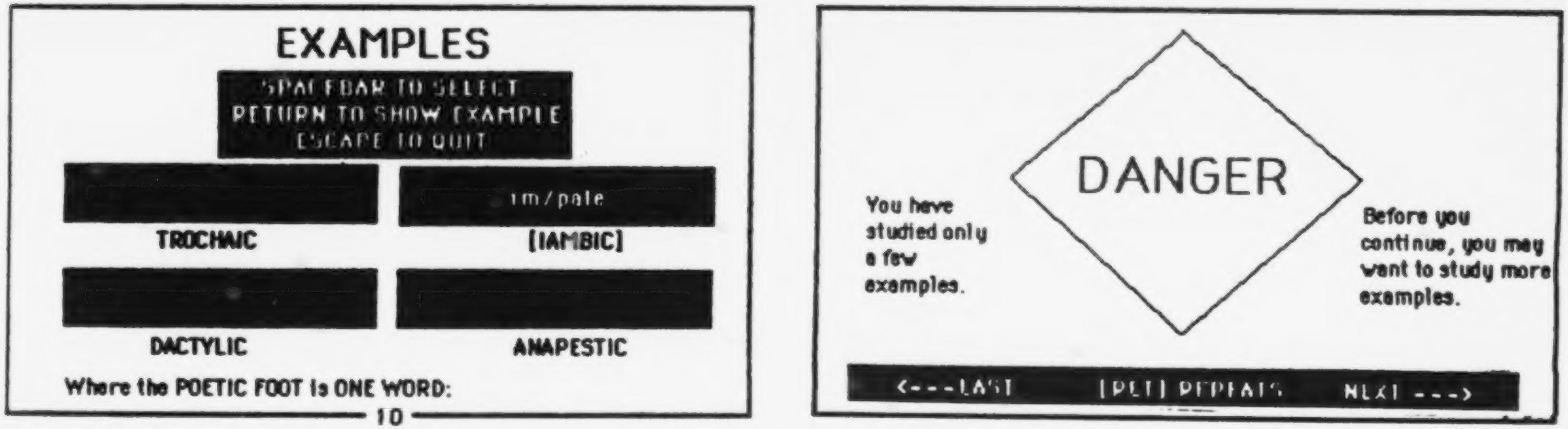
LISTEN ⇨

9

FRAME D

Figure 7 Presentation selected pages - Poetic Meter

BEST COPY AVAILABLE


EXAMPLES
SPACEBAR TO SELECT
RETURN TO SHOW EXAMPLE
ESCAPE TO QUIT
im/pale
TROCHAIC
[IAMBIC]
DACTYLIC
ANAPESTIC
10
DANGER
<---LAST
[RET] REPEATS
NEXT --->

Where the POETIC FOOT is ONE WORD:

You have studied only a few examples.

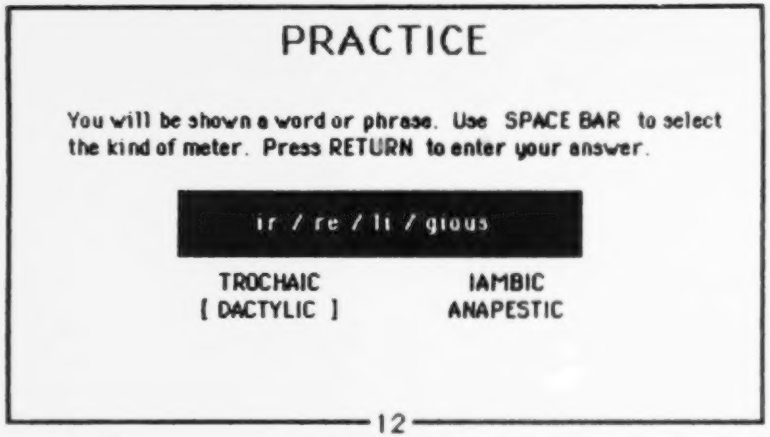
Before you continue, you may want to study more examples.

FRAME A

FRAME B

Figure 8 Word Example - Poetic Meter

BEST COPY AVAILABLE



PRACTICE

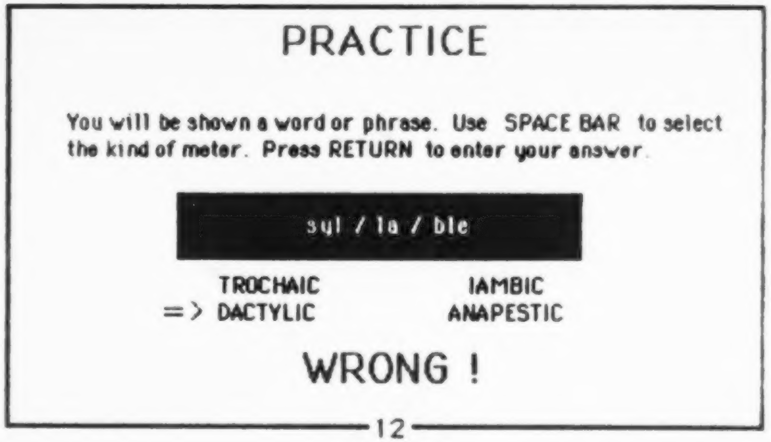
You will be shown a word or phrase. Use SPACE BAR to select the kind of meter. Press RETURN to enter your answer.

ir / re / li / gious

TROCHAIC IAMBIC
[DACTYLIC] ANAPESTIC

12

FRAME A



PRACTICE

You will be shown a word or phrase. Use SPACE BAR to select the kind of meter. Press RETURN to enter your answer.

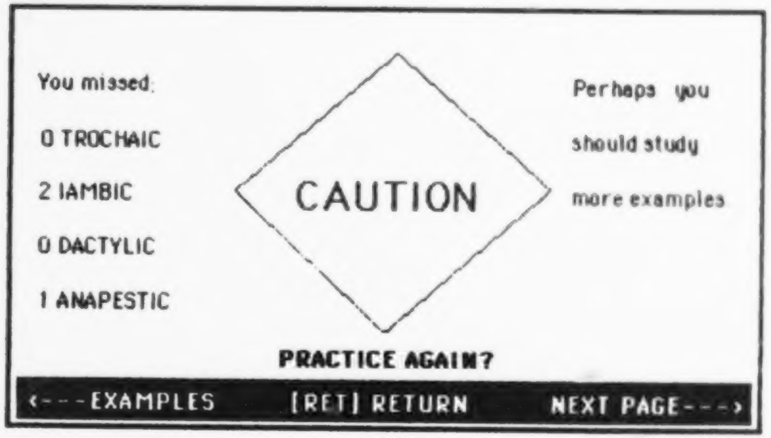
syl / la / ble

TROCHAIC IAMBIC
=> DACTYLIC ANAPESTIC

WRONG !

12

FRAME B



You missed:

0 TROCHAIC

2 IAMBIC

0 DACTYLIC

1 ANAPESTIC

CAUTION

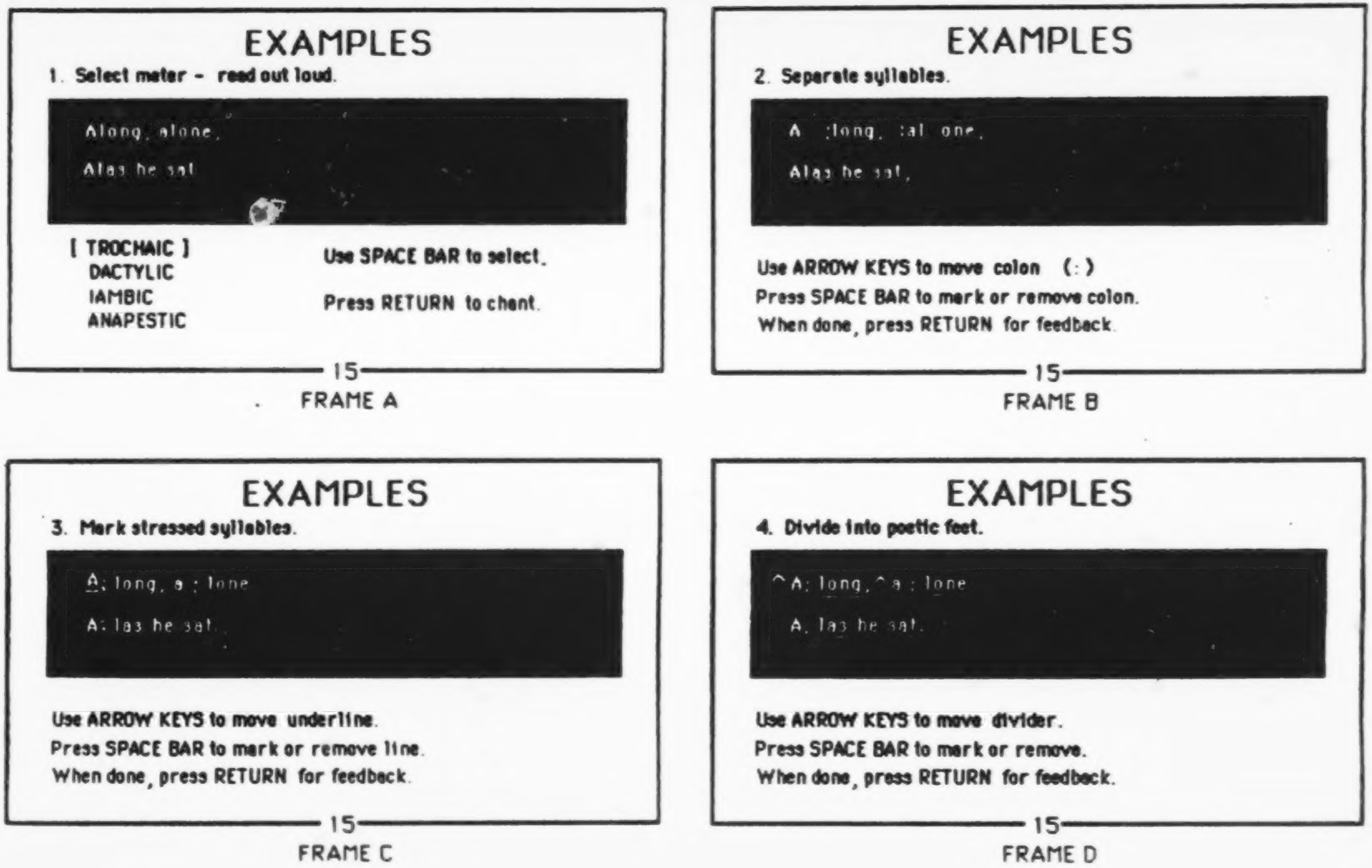
Perhaps you should study more examples

PRACTICE AGAIN?

FRAME C

Figure 9 Word Practice - Poetic Meter

BEST COPY AVAILABLE



EXAMPLES

1. Select meter - read out loud.

Along, alone,
Alas he sat

[TROCHAIC]
DACTYLIC
IAMBIC
ANAPESTIC

Use SPACE BAR to select.

Press RETURN to chant.

15

FRAME A

EXAMPLES

2. Separate syllables.

A :long, :al one,
Alas he sat,

Use ARROW KEYS to move colon (:)
Press SPACE BAR to mark or remove colon.
When done, press RETURN for feedback.

15

FRAME B

EXAMPLES

3. Mark stressed syllables.

A: long, a : lone
A: las he sat

Use ARROW KEYS to move underline.
Press SPACE BAR to mark or remove line.
When done, press RETURN for feedback.

15

FRAME C

EXAMPLES

4. Divide into poetic feet.

^A: long,^a : lone
A: las he sat.

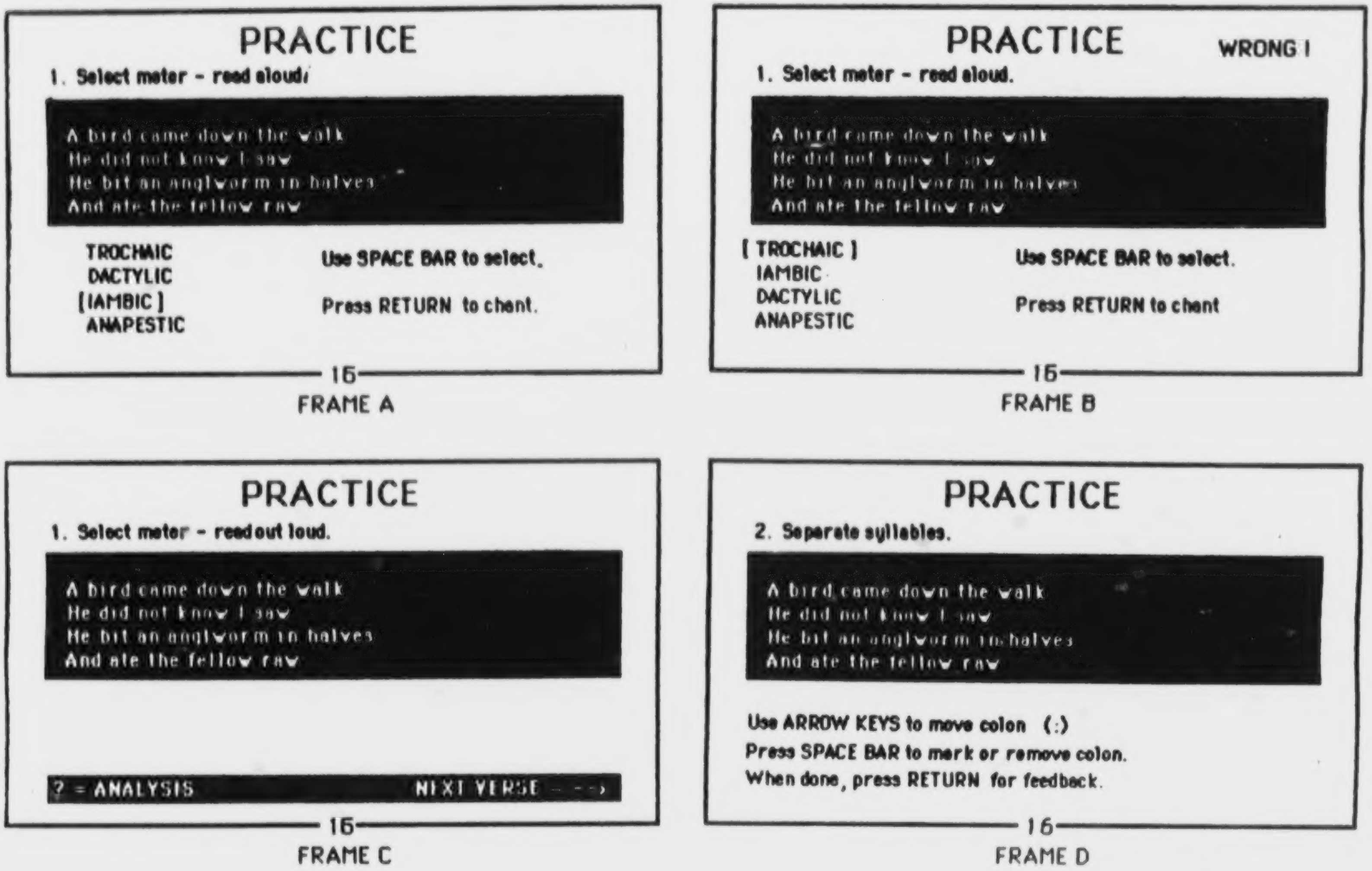
Use ARROW KEYS to move divider.
Press SPACE BAR to mark or remove.
When done, press RETURN for feedback.

15

FRAME D

Figure 10 Verse Example with scanning editor - Poetic Meter

BEST COPY AVAILABLE



PRACTICE

1. Select meter - read aloud.

```
A bird came down the walk
He did not know I saw
He bit an anglworm in halves
And ate the fellow raw
```

TROCHAIC
DACTYLIC
[IAMBIC]
ANAPESTIC

Use SPACE BAR to select.

Press RETURN to chant.

16

FRAME A

PRACTICE WRONG!

1. Select meter - read aloud.

```
A bird came down the walk
He did not know I saw
He bit an anglworm in halves
And ate the fellow raw
```

[TROCHAIC]
IAMBIC
DACTYLIC
ANAPESTIC

Use SPACE BAR to select.

Press RETURN to chant

16

FRAME B

PRACTICE

1. Select meter - read out loud.

```
A bird came down the walk
He did not know I saw
He bit an anglworm in halves
And ate the fellow raw
```

? = ANALYSIS NEXT VERSE - - >

16

FRAME C

PRACTICE

2. Separate syllables.

```
A bird came down the walk
He did not know I saw
He bit an anglworm in halves
And ate the fellow raw
```

Use ARROW KEYS to move colon (:)
Press SPACE BAR to mark or remove colon.
When done, press RETURN for feedback.

16

FRAME D

Figure 11 Verse Practice with scanning editor - Poetic Meter

BEST COPY AVAILABLE

BIOGRAPHY - M. DAVID MERRILL

Dr. M. David Merrill, Professor of Educational Psychology at the University of Southern California,and President of Microteacher, Inc., is an internationally recognized scholar in the area of computer-based instruction.

Dr. Merrill received his B.A. in 1961 from Brigham Young University and M.A. (1964) and Ph.D. (1964) from the University of Illinois in Educational and Experimental Psychology. Dr. Merrill has been Vice President/Research Scientist of Courseware, Inc. While at Brigham Young University, he served as Associate Director of the Division of Instructional Research, Development and Evaluation and Chairman of the Educational Psychology Program. In 1969, Dr. Merrill established the new department of Instructional Research and Development in the Instructional Services Division. This department and the associated Instructional Psychology Ph.D. Program which he established in 1970, attained national attention during the next decade. The department was the parent organization of the Institute for Computer Uses in Education (ICUE), and the Department of Instructional Evaluation and Testing (DIET).

Since 1963, Dr. Merrill has authored or coauthored 48 technical papers, 35 journal articles, 9 books, 10 chapters in books, 15 instructional development products, several columns and a number of reviews. He has also participated in numerous workshops and seminars.

COMPUTERIZED EXAM ADMINISTERING SYSTEM AND STUDENT GRADEBOOK

by

Bob Creel

and

Greg Baker

New Mexico State University

Introduction

The proliferation of computers in homes, businesses, and educational institutions has created many opportunities which several years ago would have been impossible. In the educational arena, computer use is not only the goal of instruction but oftentimes the means by which instruction is accomplished and enhanced. Creative use of the computer can save teachers time, make classroom presentations more interesting and effective, and increase the amount of information presented and retained. In this paper, we will present two ways in which computers have been used effectively in the educational process.

Computerized Exam Administering System

One of the time-consuming processes an instructor faces is that of constructing, administering, and grading of examinations. The first section of this paper describes a computer system developed to administer and grade examinations.

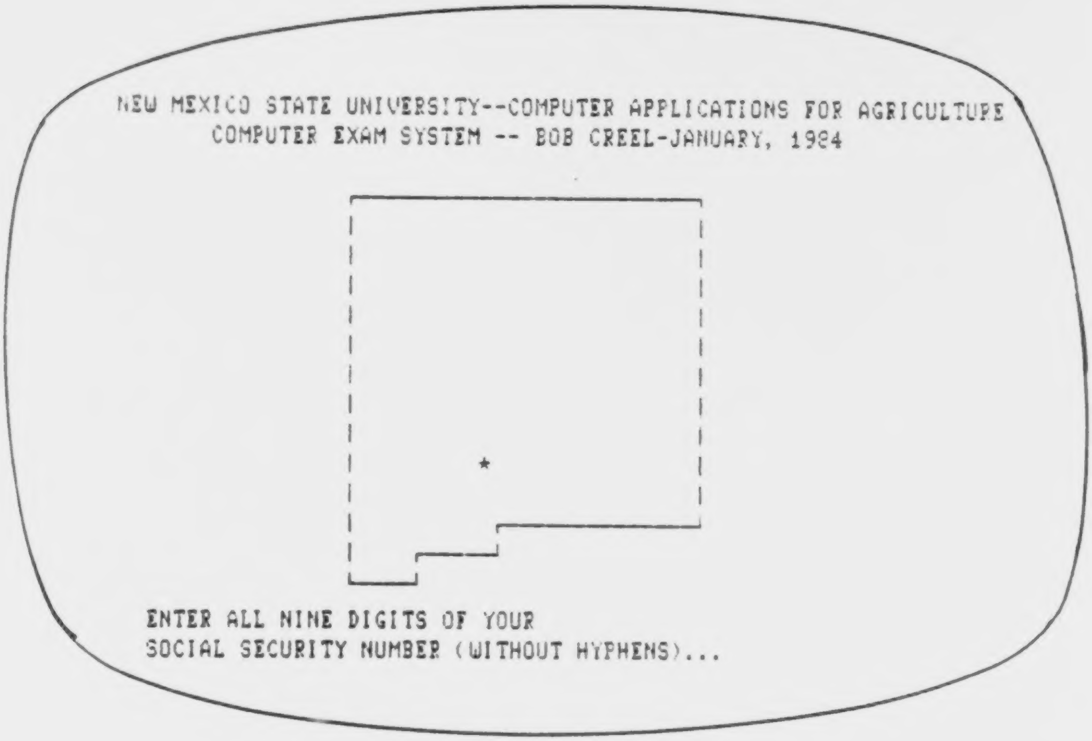
Purpose of Exams

The primary purpose of an examination is generally to measure the retention of material presented and provide the instructor with a measure of student performance. It is also desirable that examinations serve as learning tools. If students can be provided with the results of their performance promptly, it is more likely that examinations will serve as learning tools. The computer provides a vehicle to both administer the examination to the student and also provide almost instant feedbacks.

Description of Exam Administering System

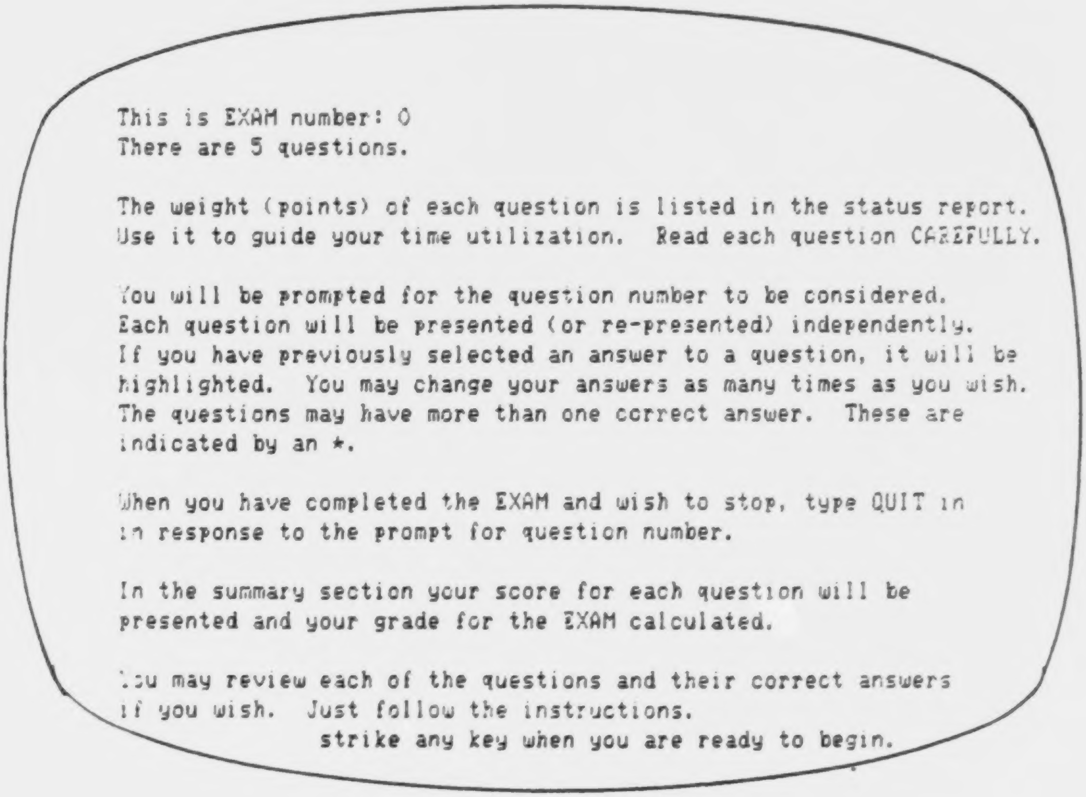
The exam administering system consists of computer programs which display a series of questions to the student, score the exam when the student has finished, allow the student to review the exam, and record the student's grade. Students gain access to the system with their social security number (figure 1). This can be used for security to prevent multiple attempts at an exam. Having accessed the system, students are presented with instructions or special information for the exam. Figure 2 shows an example of an instruction page. Multiple pages of instruction are possible. After the instructions, a status page is presented which provides the weight (points) of each question and whether or not the question has been answered or not (figure 3). Each examination can include up to 40 questions. Questions are displayed individually for consideration. The status page is represented following each question. Students can select the questions in any order, thus allowing them to address the more heavily weighted questions first. An example of a question display is shown in figure 4. The questions can be randomly selected from a bank of questions or the same set of questions can be administered to each student. The students can return to any question previously answered to check their answers as many times as they wish. The answers that they previously selected are indicated. This allows the students to review and check all their answers before they quit.

Once students have finished the exam, the system records all of their answers and scores the exam. They are presented with a status page which contains their answers to each question, the correct answer, and the score for each question as well as the overall score (figure 5).



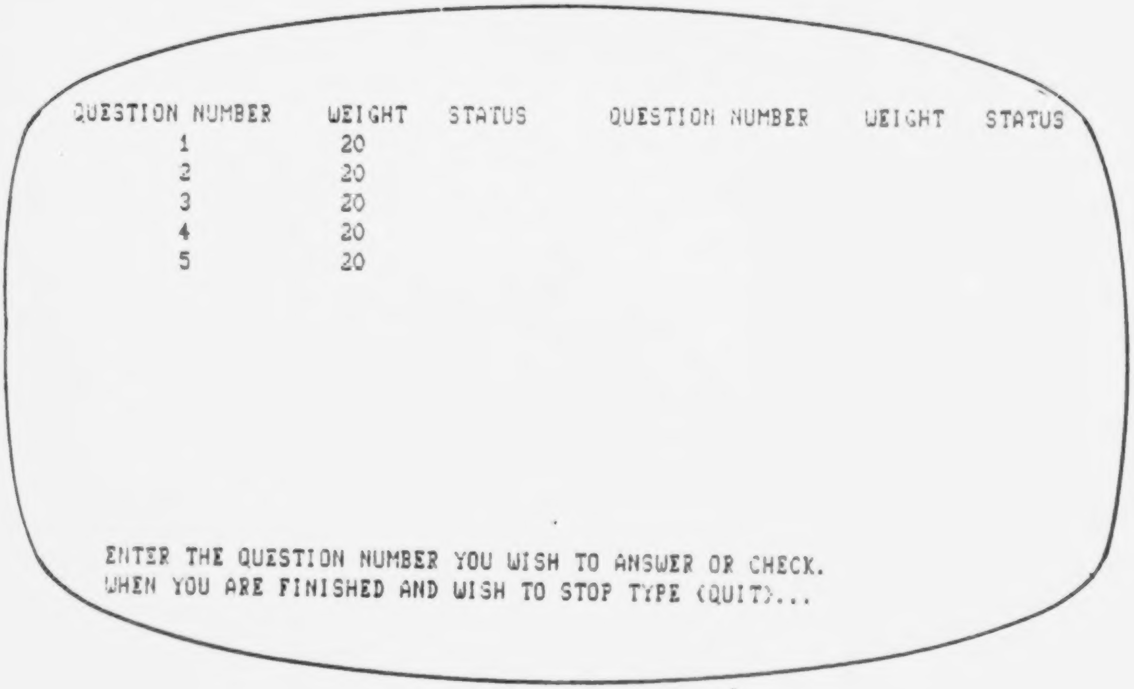
```
NEW MEXICO STATE UNIVERSITY--COMPUTER APPLICATIONS FOR AGRICULTURE
      COMPUTER EXAM SYSTEM -- BOB CREEL-JANUARY, 1984

ENTER ALL NINE DIGITS OF YOUR
SOCIAL SECURITY NUMBER (WITHOUT HYPHENS)...
```

Figure 1. Computerized Exam System screen display for access.



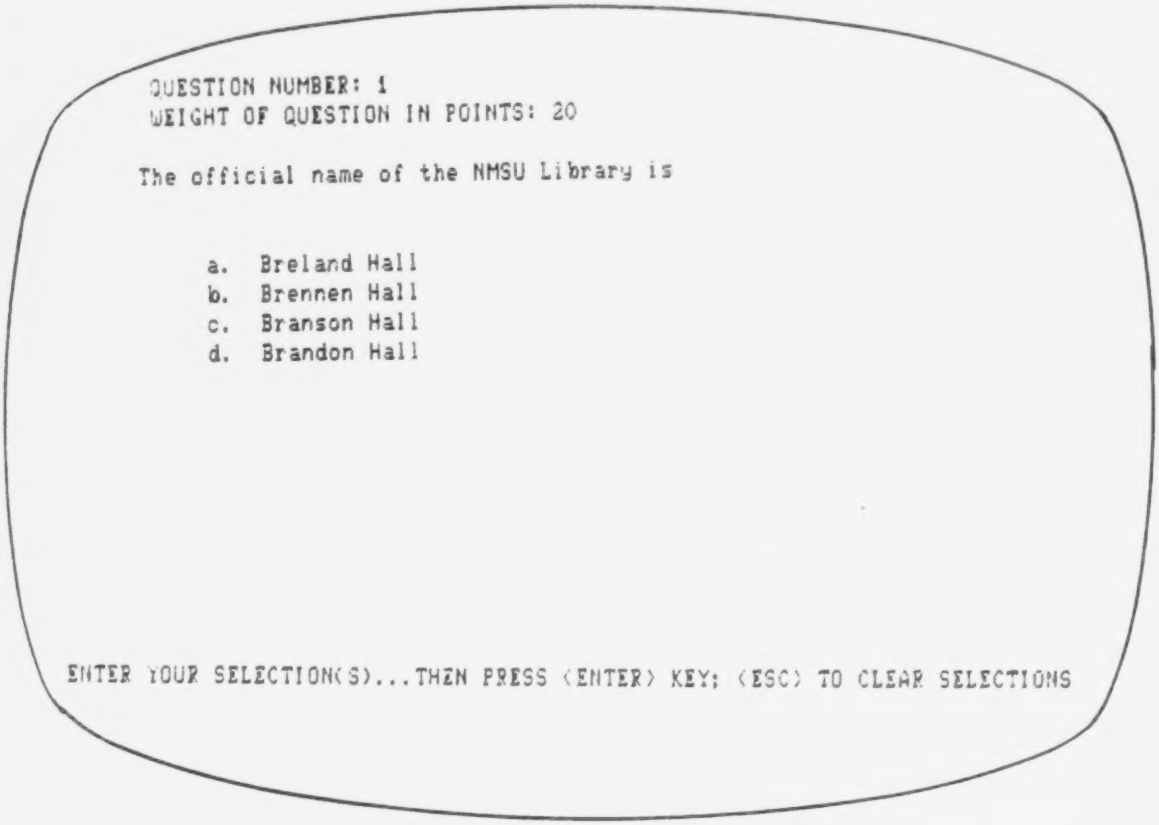
```
This is EXAM number: 0
There are 5 questions.

The weight (points) of each question is listed in the status report.
Use it to guide your time utilization.  Read each question CAREFULLY.

You will be prompted for the question number to be considered.
Each question will be presented (or re-presented) independently.
If you have previously selected an answer to a question, it will be
highlighted.  You may change your answers as many times as you wish.
The questions may have more than one correct answer.  These are
indicated by an *.

When you have completed the EXAM and wish to stop, type QUIT in
in response to the prompt for question number.

In the summary section your score for each question will be
presented and your grade for the EXAM calculated.

You may review each of the questions and their correct answers
if you wish.  Just follow the instructions.
              strike any key when you are ready to begin.
```

Figure 2. Computerized Exam System instruction page display.



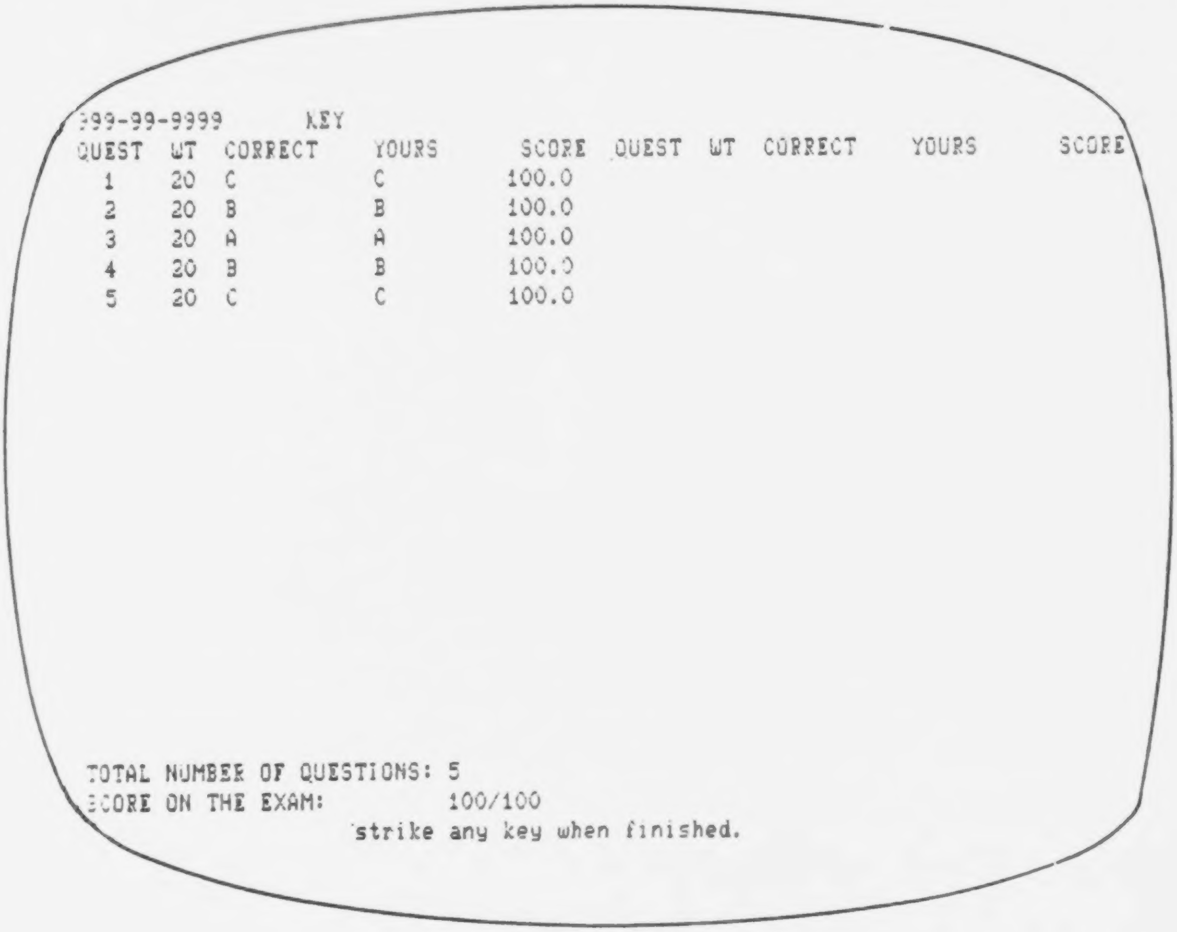
```
QUESTION NUMBER    WEIGHT   STATUS      QUESTION NUMBER    WEIGHT   STATUS
       1             20
       2             20
       3             20
       4             20
       5             20




 ENTER THE QUESTION NUMBER YOU WISH TO ANSWER OR CHECK.
 WHEN YOU ARE FINISHED AND WISH TO STOP TYPE <QUIT>...
```

Figure 3. Computerized Exam System status page display.



```
QUESTION NUMBER: 1
WEIGHT OF QUESTION IN POINTS: 20

The official name of the NMSU Library is


     a.  Breland Hall
     b.  Brennen Hall
     c.  Branson Hall
     d.  Brandon Hall




ENTER YOUR SELECTION(S)...THEN PRESS <ENTER> KEY; <ESC> TO CLEAR SELECTIONS
```

Figure 4. Computerized Exam System example question display.



```
999-99-9999       KEY
QUEST  WT  CORRECT     YOURS       SCORE  QUEST  WT  CORRECT     YOURS       SCORE
  1    20  C           C          100.0
  2    20  B           B          100.0
  3    20  A           A          100.0
  4    20  B           B          100.0
  5    20  C           C          100.0

TOTAL NUMBER OF QUESTIONS: 5
SCORE ON THE EXAM:          100/100
                    strike any key when finished.
```

Figure 5. Computerized Exam System final status page display.

Students can then elect to review the examination. The questions can be reviewed in any order the student desires. Their answers are noted as well as the correct answers. None of the answers previously provided can be changed at this point.

A number of machines can be utilized to administer the exam simultaneously. The system provides a program that compiles the information from a number of diskettes onto one file for the instructor for transfer to a gradebook.

Advantages of the Computerized Exam Administering System

The system provides a number of advantages both for the instructor as well as the student. The major advantage for the students is the prompt feedback on performance. Students are provided with their score and an opportunity to review the exam before they leave the room. The system also allows a printed version of the exam to be generated for each student with their answers and the correct answers to each question. This hard copy of the examination can be made available to students at the next class meeting.

A potential advantage for the instructor is that the instructor can devote more time to exam construction rather than to the administering and grading of exams. Additionally, scoring mistakes and biases the instructor exhibits are eliminated or reduced.

Individualized exams also reduce the potential for cheating. The system also provides a detailed analysis of the results of an exam which helps to eliminate poor questions and to identify any weaknesses in instructional presentations. The overall score for each question is provided as well as the three most prevalent wrong answers.

Limitations of a Computerized Exam Administering System

Any system for administering examinations has limitations or disadvantages. Currently, the system is limited to questions of the multiple-choice or true/false type. Essay questions cannot be scored. Start-up and learning time for the instructor to utilize the system is also required. Administering exams by computer also requires that students have ready access to machines capable of using the system. Students may feel intimidated with a computer giving them an exam, especially if they have no previous experience with a machine. The system is also subject to problems such as machine malfunction, loss of electrical power, and others.

Summary

The primary purpose of an examination is generally to measure student performance. A computerized exam administering system also provides almost instant feedback to students which allows it to serve as a learning tool. This system has been in use in both a sophomore and junior class in computer applications for three semesters. Students using the system generally like it. It has undergone a number of revisions during this time incorporating suggestions from students. The ability of students to review the exam and the time-saving features for the instructor outweigh the limitations associated with such a system.

The Computerized Student Gradebook

One of the time-consuming processes a teacher faces is that of keeping a gradebook. The use of a computer is ideal for this task. A program to store grades and to make the necessary calculations would be most useful to instructors and students.

Computer Gradebooks

There are several options available to those wanting to use computers to facilitate the grade keeping process. One type of program is the batch program such as that developed by Downey and Taylor. This system requires the user to record the grades and other data on cards or card images. The input data are then submitted to the computer along with the gradebook program. This system gives the user a great deal of control over the program, but generally requires some knowledge of the programming language.

Tice developed an interactive program for use on a micro-computer. An interactive program prompts the user with messages when input is needed. Interactive programs require a minimum of computer knowledge. The Computerized Student Gradebook (CSG) is an interactive program.

For programmers who desire the maximum in flexibility but have a limited knowledge of computers, a spreadsheet, such as Visicalc or Lotus 1-2-3, available on micro-computers is a good option (Burton). The main disadvantage of this option is that most instructors will probably not generate as much useful information for themselves or the students as with the other options.

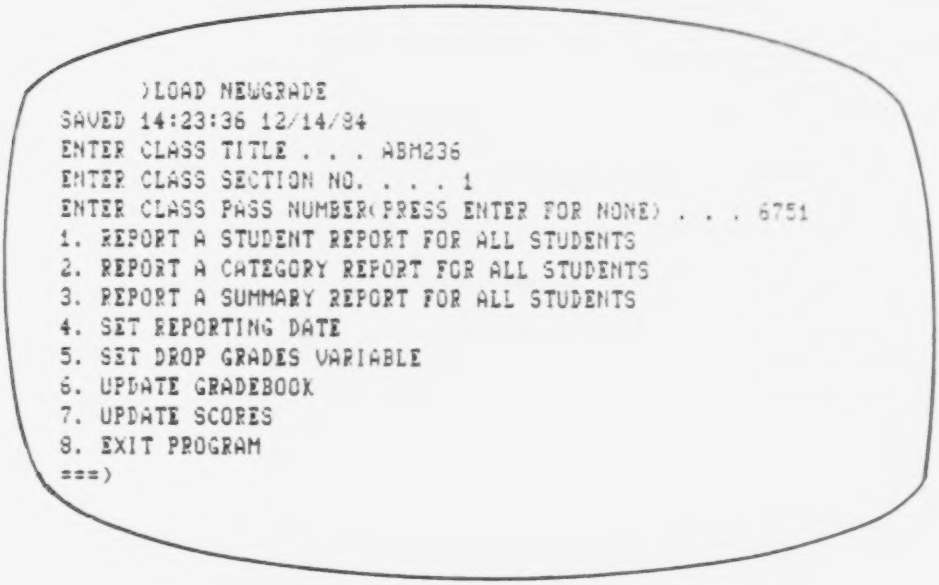
Purpose of Grade Reporting

The primary purpose of the gradebook is to record student grades. However, there are several other important functions which a gradebook reporting system should perform. For instructors, the system should make calculation of the final score as simple as possible. For an instructor who gives exams, quizzes, homeworks, laboratories, etc., the calculation of a final grade can be quite time consuming. The system should also report the final grade so that it is easy to assign a letter grade to the student or to fit the grades to a curve, if that is desired. For the student, grades should be reported often in order that the student and the instructor may monitor the student's progress. It is important, especially to students, that grades be reported accurately. And finally, students should have a good understanding of the method by which their grade is calculated. The CSG was designed with these factors in mind.

Description of the Computerized Student Gradebook

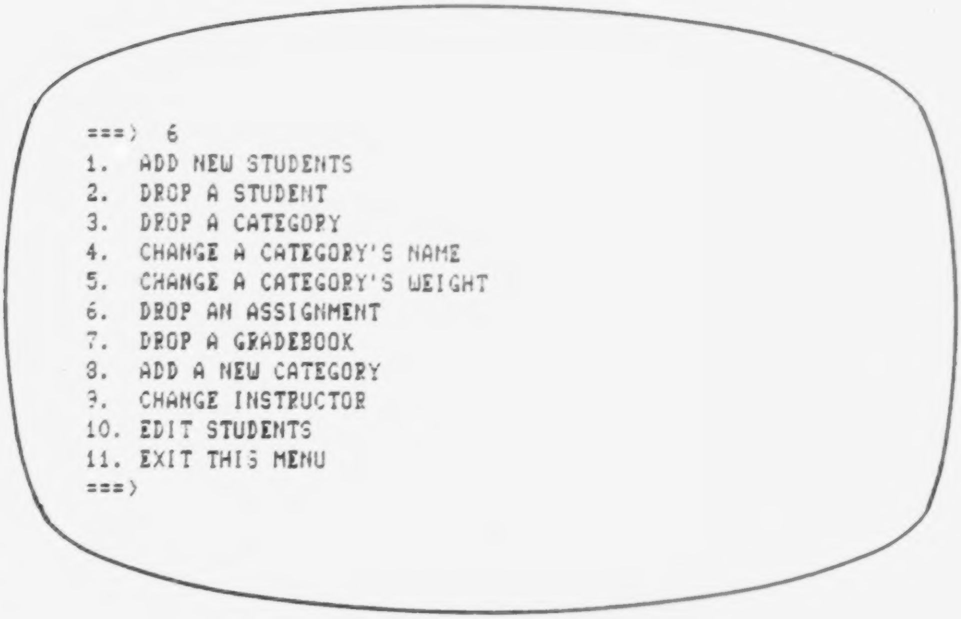
The CSG is an interactive program which can be used on either a micro or mainframe computer. Because it is an interactive program, very little knowledge of the computer is needed. The program is menu driven so that whenever input is needed from the user, a message appears on the screen informing the user of what is needed and the format for entry. This method reduces the number of data entry errors and diagnosis of the error is immediate. The main menu (figure 6) gives the user the choice of printing several reports, updating the gradebook, or updating student scores. Once the user has made a selection from the menu, another menu would appear (figure 7). This process of branching occurs until the user reaches the end of a branch and makes the desired entry, which may be changing a student's score or changing a student's name which was misspelled.

The program is designed to allow instructors to have an unlimited number of categories, such as exams, quizzes, homeworks, laboratories, and cases. The instructor assigns weights to each category, which must sum to one. The user must enter the name and social security number of each student. To enter grades, the user must specify the category for which grades will be entered and the maximum points possible for that



```
     )LOAD NEWGRADE
SAVED 14:23:36 12/14/84
ENTER CLASS TITLE . . . ABM236
ENTER CLASS SECTION NO. . . . 1
ENTER CLASS PASS NUMBER(PRESS ENTER FOR NONE) . . . 6751
1. REPORT A STUDENT REPORT FOR ALL STUDENTS
2. REPORT A CATEGORY REPORT FOR ALL STUDENTS
3. REPORT A SUMMARY REPORT FOR ALL STUDENTS
4. SET REPORTING DATE
5. SET DROP GRADES VARIABLE
6. UPDATE GRADEBOOK
7. UPDATE SCORES
8. EXIT PROGRAM
===)
```

Figure 6. Computerized gradebook system main menu display.



```
===)  6
1.  ADD NEW STUDENTS
2.  DROP A STUDENT
3.  DROP A CATEGORY
4.  CHANGE A CATEGORY'S NAME
5.  CHANGE A CATEGORY'S WEIGHT
6.  DROP AN ASSIGNMENT
7.  DROP A GRADEBOOK
8.  ADD A NEW CATEGORY
9.  CHANGE INSTRUCTOR
10. EDIT STUDENTS
11. EXIT THIS MENU
===)
```

Figure 7. Computerized gradebook system submenu display.

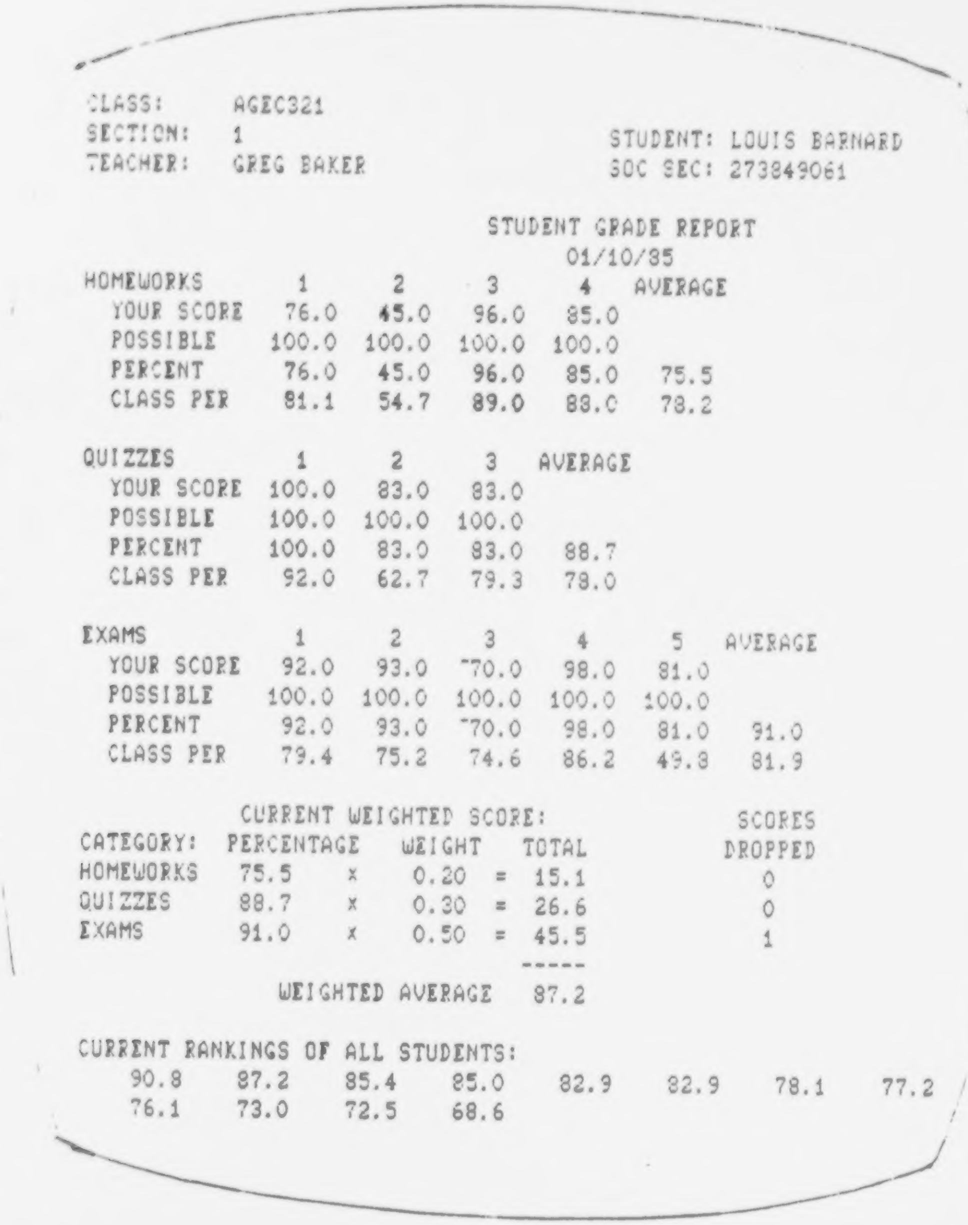
BEST COPY AVAILABLE

entry. The computer then prompts the user with a student name and the instructor enters the student's grade. The instructor may assign excused absences to a student, which means that the grade will not enter into any calculations. Anything entered by the user may be edited. This may be a student name or social security number, a category name or weight, the instructor name, or a particular score. A major time-saving feature of the program is the ability to drop the lowest grade in a category. The instructor may allow students to drop their lowest exam grade, for example. The program will then scan all exam scores and omit the lowest one from consideration in computing the exam average. This omits the time-consuming process of visually scanning the scores for the lowest percentage and recalculation of the category and weighted total average. It also reduces the number of errors which may occur when this is done by hand. The number of categories, students, or scores per student is limited only by the available storage space on the computer.

Students receive a report with the following information (figure 8):

1. All of their scores in every category, both in raw and percentage terms.
2. The class average for every score.
3. Their average for a category.
4. Their weighted total average.
5. The weighted average of all students' scores (without student identification) in descending order.

This report gives students a permanent record of every grade they have earned in the class. This helps to eliminate errors in recording the grades. Students may evaluate their progress in the course with respect to standards set forth by the instructor or in relation to the progress



```
CLASS:     AGEC321
SECTION:   1                                 STUDENT: LOUIS BARNARD
TEACHER:   GREG BAKER                        SOC SEC: 273849061

                                   STUDENT GRADE REPORT
                                         01/10/85
HOMEWORKS        1      2      3      4   AVERAGE
  YOUR SCORE   76.0   45.0   96.0   85.0
  POSSIBLE    100.0  100.0  100.0  100.0
  PERCENT      76.0   45.0   96.0   85.0   75.5
  CLASS PER    81.1   54.7   89.0   88.0   78.2

QUIZZES          1      2      3   AVERAGE
  YOUR SCORE  100.0   83.0   83.0
  POSSIBLE    100.0  100.0  100.0
  PERCENT     100.0   83.0   83.0   88.7
  CLASS PER    92.0   62.7   79.3   78.0

EXAMS            1      2      3      4      5   AVERAGE
  YOUR SCORE   92.0   93.0  ¯70.0   98.0   81.0
  POSSIBLE    100.0  100.0  100.0  100.0  100.0
  PERCENT      92.0   93.0  ¯70.0   98.0   81.0   91.0
  CLASS PER    79.4   75.2   74.6   86.2   49.8   81.9

              CURRENT WEIGHTED SCORE:                SCORES
CATEGORY:  PERCENTAGE    WEIGHT   TOTAL              DROPPED
HOMEWORKS    75.5    x    0.20  =  15.1                 0
QUIZZES      88.7    x    0.30  =  26.6                 0
EXAMS        91.0    x    0.50  =  45.5                 1
                                  -----
                WEIGHTED AVERAGE   87.2

CURRENT RANKINGS OF ALL STUDENTS:
    90.8     87.2     85.4     85.0     82.9     82.9     78.1     77.2
    76.1     73.0     72.5     68.6
```

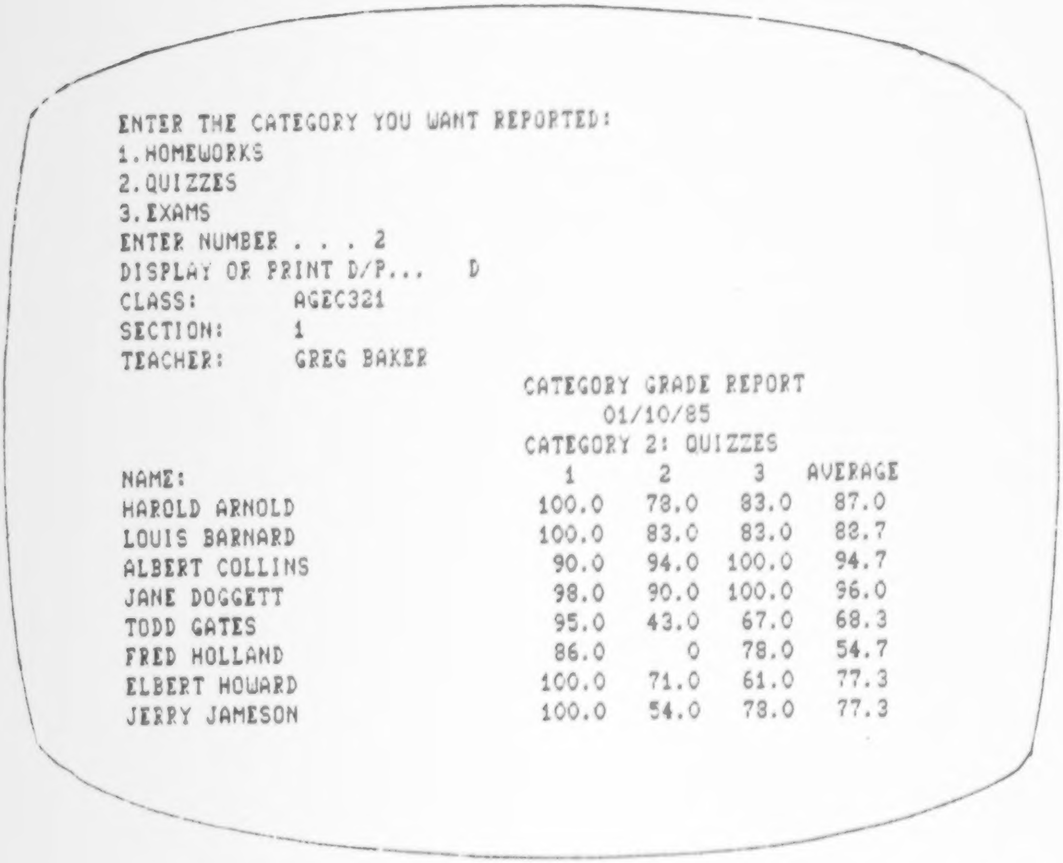
Figure 8. Computerized gradebook system student report display.

of their fellow students. Instructors may give the students this feedback at important times throughout the semester, such as before the final day to drop a class or before an important exam.

An additional feature available when using a mainframe computer is that students with access to a computer account may access their report at any time simply by loading a copy of this report. For security reasons, the students see only a copy of their report and not the original. To prevent anyone from having access to a student's report, the students must use their social security number to receive the report. In addition to allowing the instructor more flexibility as to the number of student reports to hand out, the feature may be used to encourage student use of the computer.

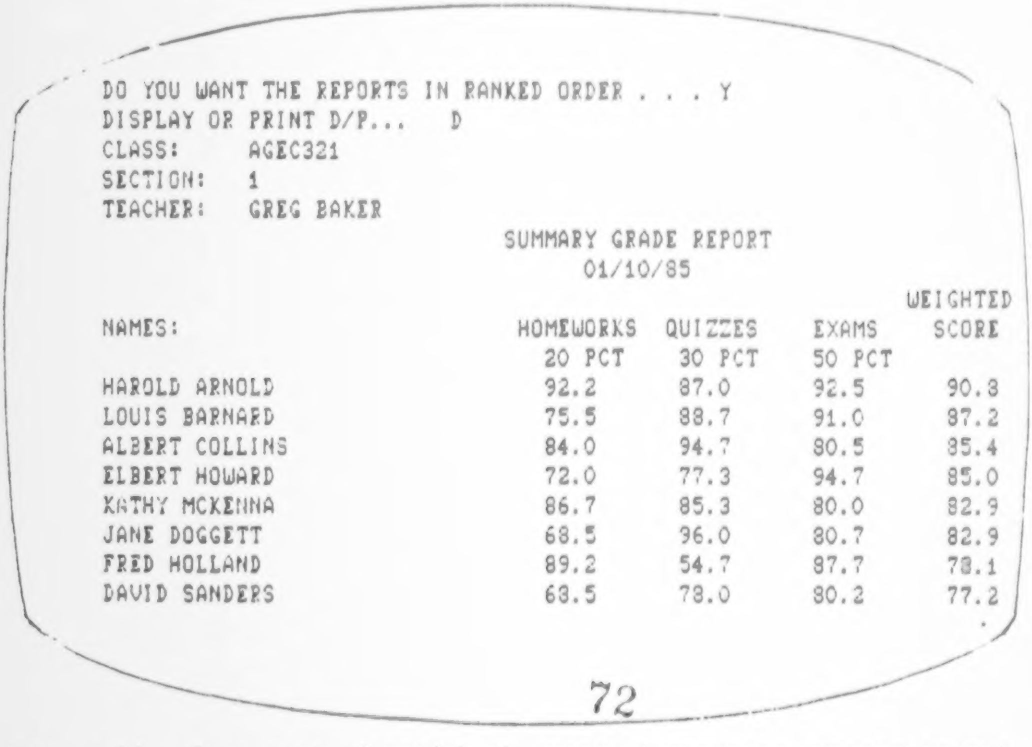
Another feature used in conjunction with computer accessed reports is a program which calculates for students (or instructors) the grade a student needs on the final exam to receive a certain grade in the course. For example, if 90 percent was the cutoff for an A, a student might ask, "What grade will I need in the last exam to receive a weighted average of 90 percent in the course?" This is time saving for both the student and the instructor.

Instructors may receive two types of reports, category reports (figure 9) and a summary report (figure 10). The category report shows every score for a student in a category and the student's average. For example, if an instructor gave three exams, the exam category report would contain the score for each student on all three exams and the average. There is a category report for each category the instructor uses.



```
ENTER THE CATEGORY YOU WANT REPORTED:
1.HOMEWORKS
2.QUIZZES
3.EXAMS
ENTER NUMBER . . . 2
DISPLAY OR PRINT D/P...   D
CLASS:        AGEC321
SECTION:      1
TEACHER:      GREG BAKER
                              CATEGORY GRADE REPORT
                                    01/10/85
                              CATEGORY 2: QUIZZES
NAME:                            1      2      3   AVERAGE
HAROLD ARNOLD                  100.0   78.0   83.0   87.0
LOUIS BARNARD                  100.0   83.0   83.0   88.7
ALBERT COLLINS                  90.0   94.0  100.0   94.7
JANE DOGGETT                    98.0   90.0  100.0   96.0
TODD GATES                      95.0   43.0   67.0   68.3
FRED HOLLAND                    86.0      0   78.0   54.7
ELBERT HOWARD                  100.0   71.0   61.0   77.3
JERRY JAMESON                  100.0   54.0   78.0   77.3
```

Figure 9. Computerized gradebook system instructor category report display.



```
DO YOU WANT THE REPORTS IN RANKED ORDER . . . Y
DISPLAY OR PRINT D/P...   D
CLASS:      AGEC321
SECTION:    1
TEACHER:    GREG BAKER
                             SUMMARY GRADE REPORT
                                   01/10/85
                                                          WEIGHTED
NAMES:                        HOMEWORKS  QUIZZES   EXAMS    SCORE
                                20 PCT    30 PCT   50 PCT
HAROLD ARNOLD                    92.2      87.0     92.5     90.8
LOUIS BARNARD                    75.5      88.7     91.0     87.2
ALBERT COLLINS                   84.0      94.7     80.5     85.4
ELBERT HOWARD                    72.0      77.3     94.7     85.0
KATHY MCKENNA                    86.7      85.3     80.0     82.9
JANE DOGGETT                     68.5      96.0     80.7     82.9
FRED HOLLAND                     89.2      54.7     87.7     78.1
DAVID SANDERS                    68.5      78.0     80.2     77.2
```

Figure 10. Computerized gradebook system instructor summary report display.

The summary report gives the instructor each student's average for a category and the student's weighted average for the course. The students may be listed in the order in which they were entered or they may be ranked by their weighted average. By ranking the students, it makes it much easier to fit the grades to a curve, since the effect of using different cutoffs may be easily observed.

Advantages of the Computerized Student Gradebook

For students, the major advantage of the CSG is that they will be better informed as to how their grades are computed and of their progress in the course. Their progress may be evaluated relative to standards set forth by the instructor or to the performance of the class as a whole, since this information is available to them. Since they have a permanent record of all of their scores, any errors made in the recording of their grades are more easily noticed. Timely use of the program and distribution of the printouts will ensure that students are aware of their performance at important times such as before class drop dates and important exams. Students also have access to their reports at any time and the system may be used to encourage student use of the computer. The program also saves time since students and instructors may use the program to determine what they need on the final exam to reach a cutoff grade in the class.

Instructors will benefit by spending less time calculating grades. They can make a great deal of information available to students and spend less time doing it than with a traditional gradebook. Two of the major time-saving features of the program are that it performs all calculations and that the lowest score (or scores) in a category may be dropped. If timely use is made of the student reports, no student should complain that he or she was misinformed of his grade. Assigning grades is much easier since all students' weighted average grades are printed out, ranked by weighted average if desired.

Disadvantages of the Computerized Student Gradebook

A disadvantage of the CSG is that some learning time is required to use the CSG. However, because the program is interactive and a system of menus is used, the little time necessary to learn the program should easily be recaptured after using it two or three times. It is also possible to lose data because of malfunctions in hardware or because of the system going down unexpectedly. A computerized gradebook is not a substitute for the traditional gradebook, but can be a very useful supplement. As a matter of security, both should be used. Another problem is that when using a mainframe computer, one has little control over use of the computer or printer. And, although the CSG was written to allow the maximum in flexibility, the user must still operate within the limits of the program and the computer.

Summary

One of the time-consuming tasks that an instructor faces is that of keeping a gradebook. A number of computerized gradebook programs are available, but these have a number of limitations. The CSG system provides a user-friendly environment with many options unavailable in other systems. The benefits provided with the CSG system outweigh the costs associated with its use.

References

Burton, Robert O., Jr, "A Computerized Gradebook: One Teacher's Experience," National Association of Colleges and Teachers of Agriculture Journal, June 1984.

Downey, W. David, and Robert W. Taylor, Student Grade Report, unpublished paper.

Tice, Thomas F., "Grade Recording and Statistics Program for Microcomputers," National Association of Colleges and Teachers of Agriculture Journal, December 1981.

A RETESTING ON DEMAND STRATEGY FOR GENERAL PHYSICS USING COMPUTER GENERATED PROBLEMS

by

A. F. Burr

New Mexico State University

A Retesting on Demand Strategy for General Physics Using Computer Generated Problems

A. F. Burr

Physics Department
New Mexico State University

This report will be divided into two parts. The first part will be concerned with the educational technique of multiple retesting. The second part will be concerned with the computer program and data base which makes the retesting technique feasible. This report and the whole project is of a practical rather than theoretical nature. The purpose of the project was to encourage learning rather than to investigate or develop some educational theory.

The course directly involved is a conventional one year non-calculus general physics course. It provides a standard treatment of mechanics, heat, sound, electricity, magnetism, light, and modern physics. The course is divided into two semesters of three credits each. There is a one credit hour per semester laboratory associated with, but not a part of, the course. The ideas and findings developed during this project should be just as applicable to any similar course. That is, the mathematical level, the subject matter level, or even the subject matter itself does not play an important role in the study. Any course which could use a test of the type used in this study would be suitable for the techniques described here.

The Grading Scheme Used for the Course

Grades were assigned at the end of the semester in such a way that the average was a high C and the distribution approximately normal. Three factors made up the final marks: homework (10%), three hour tests (70%), and a one hour comprehensive final test (20%). All components of the final grade are similar. The homework consists of standard problems selected from the problems given at the end of the Chapters of the text. During the semester, three hour tests were given at approximately four week intervals. These tests covered only the material studied since the previous test. A comprehensive final was administered at the last class meeting of the semester. The questions on all the tests were all of the same type as the homework problems. They contained no true or false, multiple choice, or short answer type of questions.

The retesting program basically offered to any student the opportunity to take another test over the same material as any previous hour test. These retests could be taken at any convenient time. As many retests as the student wanted were offered.

Retesting as an Educational Strategy

Some people ask why should one test at all. Most teachers consider tests to be a necessary, if not always a good, feature of most courses. Three often used reasons for testing are

1. A direct and immediate incentive for the student to study.

2. An opportunity to integrate and organize the knowledge gained from the course.

3. To provide a basis for grade assignment.

Retesting

Retesting over the same material is a technique which is a little more controversial than testing itself. A number of arguments are put forward in favor of retesting over old material.[1,2] The two main ones are greater mastery of the subject matter by the students and a more relaxed attitude on the part of the students. The students also have a greater incentive to study. The retesting process also provides for automatic make-up tests for those students who are absent. Students often claim they have not been given enough points or that the test question selection was not fair. These students can be simply told that, if for any reason they think their grade does not fairly represent their knowledge, they can ask for a retest.

There are some legitimate arguments against retesting. One of the most telling is that it is more work for the instructor. It also increases the chance of system manipulation by the student. It forces another choice on the student and may reduce the seriousness with which the student regards the regular tests. Some people[3] object that the procedure allows students of marginal intellect to get higher grades and even some students object to the curve raising aspects of the strategy.

This experience has shown me that students do study more and learn more. The extra work can be manageable also. Students do try to manipulate the system to their advantage. An unexpected bonus was the almost entire disappearance of some problems. I no longer had to worry about doctor's excuses, funerals for dead aunts, or other possibly fabricated excuses for missed tests. Neither did I have to argue about the sufficiency of part credit for problems not completely botched. The fairness of problem selection could no longer be effectively questioned. The one all purpose response to all the difficulties was "At what time would you like to take the retest?"

Details

There are a number of details which have to be considered if any program of extensive retesting is to be successful. The first to be settled is the manner in which retest scores are to be counted. Discussed here are four ways of handling this problem. They are listed in my order of preference.

I prefer to record the last test score. Of course this procedure means that a student will not improve his class standing if his retest score is lower than his original score. This element of risk reduces frivolous retesting and reduce the tendency to take as many tests as possible in the hopes that one lucks out on one and gets a high score.

Strictly speaking a good case can be made for the use of the average of the original plus all the retests. In this way you make use of all the information available. You prevent the student from underestimating the importance of the first test. If, after a disastrously low first score, the material is truly learned so that high scores are routinely obtained, the final score can still be raised to any desired value by amassing a number of high scores to offset the original low one. From a practical standpoint, however, it is a little selfish to demand that much time of a student at the expense of his other courses.

Some teachers feel that they are being generous if they award 50% of the improvement generated by retests. There are some philosophical difficulties with this procedure. One hopes that the tests measure the knowledge the students have of the physics course. The purpose of the course is to teach physics. The course is not an intelligence test or a screening device for various occupations or other courses. If a student passes a test showing he now knows the material he should not be penalized for his previous ignorance.

The students would prefer to be awarded the points for the highest score made regardless of areas of ignorance uncovered by subsequent tests. That method leads to too many students taking tests on speculation. It increases the workload and decreases the built-in incentives for study before the test.

The question of part score for problems incorrectly solved because of a minor arithmetic or other error is a serious one. With all the retests given and the different problems posed, there just is not enough time to locate the error which a student has made. A good case can be made for the all or nothing approach to the grading of problems. If one is designing a bridge and one comes up with a good design except for the fact that the decimal point is off by only one place, you are still going to get fired, and sued, when the first truck across it falls through. Nevertheless I find it difficult not to give part credit so I throw in a few points for the correct units or the prominently displayed correct physical principle. I have to force myself not to try to locate the students error because I would have to steal the time required from other activities which the University, my department head, and I consider important. This problem is serious, however, and is the biggest cause of student dissatisfaction with the retesting procedure.

There are a number of housekeeping problems. One has to find places to set the students while they take the retest. To put many in the same room would require supervision. The constant interruption of other tasks by students asking for retests decreases efficiency. Many of these problems are solved by asking the students to sign up for one of six specific times during the day. Then the number at a time is limited to the number of testing places available. To allow time for study, a maximum of one retest per week is permitted. The tests are never graded immediately and the student must make a demonstratable effort to find his or her error before the answers are discussed with them individually.

Cheating, especially in the form of a small crib sheet containing equations, is a possibility because all these retests cannot be supervised individually. An attempt to look in on each student at some random time is made but a student determined to cheat could. All work is required to be turned in with the answer. Few suspiciously high scores have been seen.

Results

The program of multiple retests has been used for three semesters. Table 1 summarizes some of the results that have been obtained.

The interpretation of these statistics is obscured a bit by the fact that for two of these semesters the last test score was the one counted while for the other semester it was the average which was used. With the latter method of scoring the number of students requesting retests, the number of retests given, and the improvement in test scores were all about one half of the corresponding statistic for the semester in which the last test was the one which counted. These results are consistent with results reported from somewhat similar projects.[4]

All the students were encouraged, but not required, to comment on the retesting procedures with the results shown in Table II. The students were very enthusiastic about the program. There were comments like "Great Idea!", "best part of the course". Of those who explained their dislike of the program, two said they had no time for the retests and five said they resented the fact that those with more time to study could improve their relative class ranking at the expense of those without the time.

My subjective observations make me an enthusiastic supporter of the program. The amount of time spent studying physics clearly increased. The number of intelligent questions asked in and after class increased. Fewer students gave up trying to master earlier concepts, and the number of arguments over test grading decreased.

The Computerized Tests

The availability of computer generated tests is the feature which made the retesting program possible. Without them, it would have been impossible to find the time to create and grade the tests. A computer program was written which randomly selected a prescribed number of questions (usually four) from a prescribed portion of a previously entered data base. The questions were assembled into a test complete with a unique serial number and a place to write the answer.

Each of these questions consisted of a problem, for which there was usually a numerical answer, of the same type as were given for homework. The numerical value of one or more of the variables in the problems was selected at random by the computer from a previously designated range. Thus even with a small data base there was no possibility of two identical tests.

The computer also calculated the answer to each particular problem and printed it, complete with identification and appropriate units, right beside the space which it provided for the student's calculated answer. The list of answers is printed on a part of the page separated from the test by perforations. The answer list is also identified by the same serial number assigned to the test. Figure 1 shows part of a typical test.

Just before the test is given to the student, the answers are torn off. Then at the end of the day all one has to do is match the tests to the answer sheets and compare the response of the student with the results provided by the computer. Time does not permit the details of the work which the student is required to turn in with the test to be checked for part credit. Often, though, a small bit of credit can be quickly given for correct units.

The Data Base

The data base was created by typing into a special file in the computer problems from the back of the chapters of the text.

Although the project is not yet complete, enough problems have been entered to provide a reasonable selection for each chapter. Figure 2 shows a typical problem as entered into the data base. Each problem has the same six parts as shown in the figure.

The first part is the problem identifier. The first number (here 02) is the chapter number. The second (here 01) is the section number. The third (here 2) is a serial number to ensure uniqueness. The three numbers which follow (here 2 1 1) denote, respectively, the number of variables for which the computer must supply random numbers, the number of parts to the question, and a number proportional to the estimated difficulty of the problem. The last number is a 1,2, or 3.

The next section consists of a line for each variable. The first entry is the symbol for the variable. The first of the string of numbers following this symbol is the lower limit of the range from which the random number is to be drawn. The next number is the upper limit of that range. The third number is the power of ten to which the random number is to be raised, that is it determines the location of the decimal point. The other three numbers are provided automatically by the computer and are used in the internal program.

The following section (set off from the rest of the problem by semi-colons) is the equation used by the computer to calculate the answer to the problem after it has selected all the required random numbers. This equation must follow rules similar to any instruction in FORTRAN or BASIC.

The next line contains whatever is to be printed in front of the numerical value on the answer sheet.

The next line contains whatever is to be printed after this numerical value. Usually it is the units of the answer.

The final part is the text of the problem itself. In this text the variables which are to be replaced with random numbers are represented by the symbol previously defined preceeded by an ampersand.

The Entry Program

A separate program was written to assist the person entering these parts of each problem into the data base. The keyboard operator is prompted to enter each of the parts discussed above. The operator has the option to enter the entire section at the first prompt. If this option is declined, the computer will prompt for each part of the section. A limited amount of error checking is provided, mostly consisting of recycling through the prompts if obviously inappropriate data is entered.

An overall check for correctness is provided. When the details of the random variables are entered, one of the prompts requests a sample value for the variable. After the equation is entered, the computer uses these sample values to calculate an answer. This answer is displayed and the operator is asked if it is correct or not. If it is, the computer goes on to the next operation. If it is not, the operator is given the choice of rewriting the equation or of abandoning this particular problem.

Because it takes some computing time to add problems to the data base the usual procedure is to collect new problems in a temporary file until a convenient time. Then the computer is instructed to add the temporary file to the main data base. The temporary file can be easily checked for errors and edited before being merged with the main file.

The Test Making Program

There is another program for actually making the tests. This program is also self-prompting so that persons without an extensive computer background can create tests easily. After telling the computer that one wishes some tests, the computer asks if instructions on running the test making program are desired. It then provides them or asks a series of questions which define the tests to be created. These questions are

Do you wish to:

1. Use a set of characteristics that you have previously saved?
2. Specify a new set of characteristics?

Enter the number of tests to be generated:

Enter the number of questions to be included in each test:

Do you want

1. To have the SAME questions on each test?
2. To have DIFFERENT questions on each test?

Enter the chapters to be covered on the test:

Chapter number ...

Enter any sections which MUST be included on the tests:

Enter any sections which MUST NOT be included on the tests:

Enter any questions which MUST be included on the tests:

Enter any questions which MUST NOT be included on the tests:

The answers to these questions can be filed in a test characteristic file for later use if desired.

Difficulties

There are a number of difficulties in creating the data base. The most important one is the difficulty of including even the simplist graphics. In practice this means that any problem which depends upon a figure for its solution cannot be included in the data base. Furthermore any problem which requires a figure for its answer will be more difficult to grade. Indeed most problems which cannot be answered with a numerical or short, well defined answer will be too time consuming to grade.

Another problem is raised by variables which are to be randomized. At first it was planned to randomize every numerical value which appeared in a problem; but this plan proved too time consuming (and unnecessary besides). First of all one must limit the range from which the computer selects a number for a variable. If this is not done, the correct answer to a problem may be unphysical in magnitude. Because every instructor would like the student to at least make sure that his answer is reasonable, it would be self-defeating to have the correct answer to the speed of a runner to be something like 127 meters per second. Thus if the question is something like "What is the average speed of a runner who covers 100 meters in 10.3 seconds?" cannot be randomized without thought. In this case one might ask the computer to select the time from the range 10 to 15. If both the time and the distance were to be randomized, somewhat more than twice the effort spent on radomizing one would have to be spent to keep the answer within reason. Furthermore each variable randomized required its own line in the entry program. Fortunately, in most cases it is only required to randomize one, or at most two, variables to change the problem enough to prevent students from getting credit for only memorizing a specific answer. Of course some problems, such as the addition of a number of vectors can have more variables easily randomized.

Yet another difficulty is caused by the requirement that the answer to each problem (or part of a problem) be represented by a single equation in the variables randomized. This equation has to be found, usually an easy task but sometimes time consuming. For example, when solving for the direction of the resultant of a sum of vectors, one usually takes the arc tan of the ratio of two previously calculated simple numbers, the sum of the x and y components. Here one cannot reduce the computer to numbers but must write them out as algebraic sums; a procedure which results in a long and unweilding acr tan equation.

For this project, a file of physical constants, such as the charge on the electron, was created so that they could be referred to be letters. It turned out that the complications introduced by the additional file were more than the convenience of not having to explicitly enter the constants into the equation for the problems.

Another problem was how to get the computer to print out powers of 10. The usual printer will not handle numbers such as 3.6×10^6 meters. Various printed forms such as 3.6x10E6 or 3.6E6 were tried. The latter form was selected because it is the usual form printed by computers and it proved to be easy to learn and recognize even by students with no previous computer experience.

In all the program has turned out rather successfully. It proved to be more difficult than expected to create the data base, but the tests are very easy to create and convenient to grade. The program could also be used to create new home work problems, each set to be slightly different from the other to prevent one answer from being used by several students.

References

1. David E. Golden, Robert G. Fuller, and Donald D. Jensen, Am. J. Phys. 42, 941-3 (1974).
2. David E. Golden, Robert G. Fuller, and Donald D. Jensen, Am. J. Phys. 43, 746 (1975).
3. John Lohr, Am. J. Phys.43, 738-9 (1975).
4. James D. Edmonds, Am. J. Phys. 43, 745-6 (1975).

Table I. Retesting Statistics

Number of Students who could request retests	177
Number of Students Requesting Retests	149
Number of Retests Given	512
Retests per Student	29
Average Score Improvement	27
Average Percentage Increase	15

Table II. Student Comments

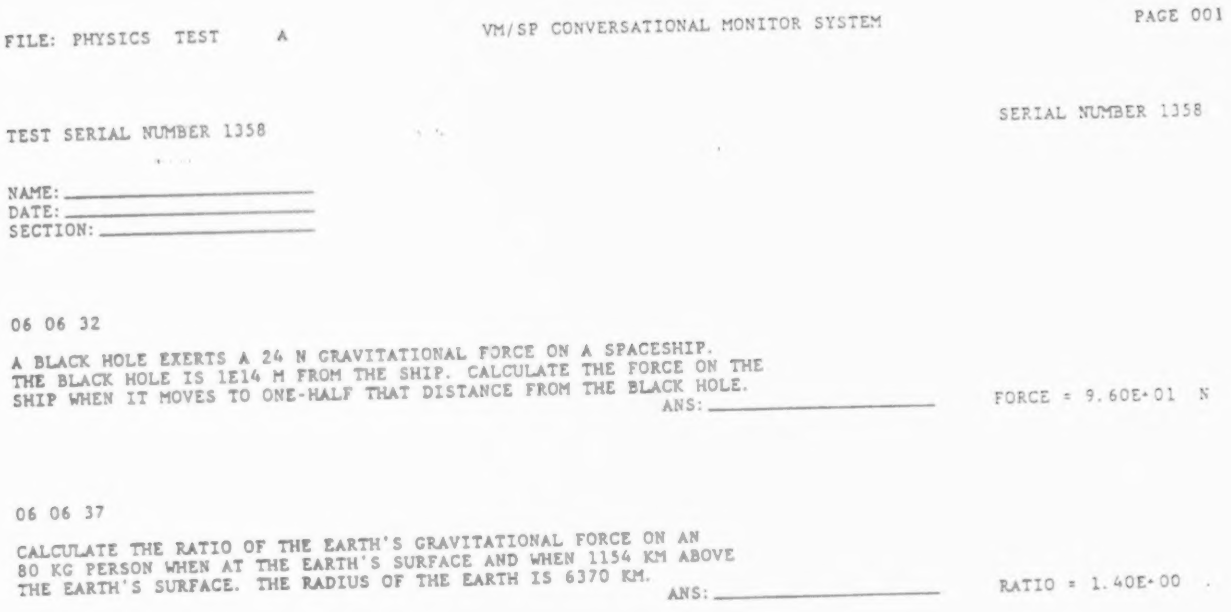
Did you like the retesting program?

Yes	91%
No	9%

Why?

A.	The retests were easier	8
B.	Was able to study more	51
C.	Felt less pressure	42
D.	Was able to guess which questions would asked	24
E.	Other	4

(Multiple answers permitted)



```
FILE: PHYSICS  TEST     A                VM/SP CONVERSATIONAL MONITOR SYSTEM                PAGE 001


TEST SERIAL NUMBER 1358                                                     SERIAL NUMBER 1358

NAME: ______________________
DATE: ______________________
SECTION: ___________________


06 06 32

A BLACK HOLE EXERTS A 24 N GRAVITATIONAL FORCE ON A SPACESHIP.
THE BLACK HOLE IS 1E14 M FROM THE SHIP. CALCULATE THE FORCE ON THE
SHIP WHEN IT MOVES TO ONE-HALF THAT DISTANCE FROM THE BLACK HOLE.
                                             ANS: ________________      FORCE = 9.60E+01  N


06 06 37

CALCULATE THE RATIO OF THE EARTH'S GRAVITATIONAL FORCE ON AN
80 KG PERSON WHEN AT THE EARTH'S SURFACE AND WHEN 1154 KM ABOVE
THE EARTH'S SURFACE. THE RADIUS OF THE EARTH IS 6370 KM.
                                             ANS: ________________      RATIO = 1.40E+00
```

Fig. 1. Part of a typical test with the answer on a perforated section to the right.

```
02 01 2    2 1 1
        T1      2 9 0 1 0 2
        T2      2 9 0 1 0 4
      ;
         SQRT((T1 * 0.799 + T2 * 0.174) * (0.799 * T1 + T2 * 0.174) +
         (T2 * 0.985  T1 * 0.602) * (0.985 * T2 - 0.602 * T1))
      ;
        FORCE -
        N
          A FORCE TABLE USED IN STUDENT LABORATORIES IN PHYSICS CLASSES
        HAS A RING TO WHICH THREE STRINGS ARE TIED.  THE FIRST TWO STRINGS
        EXERT THE FOLLOWING FORCES ON THE RING: T1 (&T1 N DIRECTED 37 DEGREES
        BELOW THE POSITIVE X AXIS) AND T2 (&T2 N DIRECTED 80 DEGREES ABOVE
        THE POSITIVE X AXIS.)  CALCULATE THE MAGNITUDE OF THE FORCE OF THE
        THIRD STRING ON THE RING IF IT REMAINS IN EQUILIBRIUM.
```

Fig. 2. Typical problem from the data base.

COMPUTER BASED SPECIAL EDUCATION TEACHER TRAINING

by

G. Phillip Cartwright

The Pennsylvania State University

96

COMPUTER BASED SPECIAL EDUCATION
TEACHER TRAINING

G. Phillip Cartwright, Ph.D.
Professor and Head,
Division of Special Education and Communication Disorders
The Pennsylvania State University

Problem 1:

The Pennsylvania State University is a multi-campus institution with 20 Commonwealth or branch campuses in addition to the main campus. Most students attend a branch campus for two years then transfer to the main campus for the last two years of their baccalaureate programs. Our problem: there are no Special Education faculty at the branch campuses but there is a real need to begin instruction about problems of educating the handicapped before the junior year of a 4-year program.

Problem 2:

In 1980, the State Department of Education mandated that all persons who apply for teaching and instructional certificates in any area of education (e.g., counseling, teaching chemistry, supervision, elementary education) must demonstrate ten generic competencies dealing with education of the handicapped. A similar problem existed: insufficient staff at the branch campuses and the main campus to provide even modestly individualized programs or courses of study for such diverse groups.

Problem 3:

There is a need to provide a high quality computer based correspondence course to teachers and parents who want basic information about handicapped children.

Our Solution:

In response to the needs sketched above, the University has developed a series of microcomputer modules and an entire three credit course to assist University students, parents, and inservice teachers to obtain basic information about the handicapped.

In addition to the main campus, the University maintains 20 branch campuses which offer basic freshman and sophomore courses. No Special Education faculty are employed at any of the branch campuses. In the past, students in many majors had to wait until their junior year to obtain instruction dealing with education of the handicapped. Now, however, students at any of the campuses can take a three credit course in Special Education at any time. Similarly, students, parents, or teachers in rural areas can take a full course or a set of modules on their own if they have access to a computer.

An Introduction to the Education of the Handicapped is given by microcomputer and is a "stand-alone" course; all instruction is given by the computer through a series of computer modules. The modules are easily transportable to other institutions and use the following computers: Apple (II+, IIc, IIe); TRS-80 (Models III and IV); IBM (PC, PC XT, PCjr).

The course, or a selected set of modules prescribed for the individual student, also prepares students for the special education generic competency examination, which is required University-wide.

The components of the course are: (a) a set of 19 floppy disks (depending upon computer system); (b) a printed manual which provides very brief summaries of each of the lessons on the disks, chapter overviews, references, and related information; (c) an instructional program on the use of the computer, (d) instructor's manual; and, (e) evaluation packet containing quizzes and final examination.

Students are expected to read a chapter in the textbook first, then study the same chapter on the computer. The material is self-instructional and no prior knowledge of computers is necessary. Use of the computer is taught on the first disk. Students are on their own to pursue the materials and may do so at their own pace. However, students should plan to schedule 3 to 5 hours per week computer time.

Subsequent material is completely self-instructional and self-paced. Each book chapter has a corresponding chapter which is programmed onto one to three disks. Each disk takes 30 to 60 minutes to complete, and the estimated instruction time spent on the computer equals roughly 37½ clock hours, the same amount of time as is required for a conventional three credit course. Because the program is self-paced, some students may complete the lessons quickly, where others may work at a slower pace and repeat the lessons as often as they wish.

The course has been in use at Penn State for 2 years; approximately 300 students have completed the course. The entire program is available from Penn State's Department of Audio-Visual Services. Several universities and school systems in the US and Canada have obtained the material for use within their own institutions.

GPC4/P6

SOME USES OF A COMPUTER IN TEACHING, TRAINING, AND TESTING: STUDENT/LEARNER SELF-ASSESSMENT RESPONDING

by

Darwin P. Hunt
New Mexico State University

SOME USES OF A COMPUTER IN TEACHING, TRAINING, AND TESTING
Student/Learner Self-Assessment Responding

Darwin P. Hunt
Experimental Training Center
Department of Psychology
New Mexico State University
10 December 1984

> "The proud man...is an extreme in respect of the greatness of his claims, but a mean in respect of the rightness of them; for he claims what is in accordance with his merits, while others go to excess or fall short."
>
> "...he who thinks himself worthy of great things, being unworthy of them is vain."
>
> Aristotle, 4th century B.C.

ABSTRACT

When considering computers as a machine component in teaching, training and testing it is useful to employ systems theory as a conceptual framework. Skinner's proposal that the label, 'computer,' is inaccurate and counterproductive when used to refer to a machine being used for teaching is discussed. An alternative label, 'teaching machine,' has the disadvantage of being associated with the past failure of "teaching machines" to live up to their expectations. Three practical applications based upon our own research concerning the human "self-assessment" process are discussed. They are (a) self-assessment multiple choice tests, (b) self-assessment computer aided testing and (c) self-assessment computer aided instruction and review. The underlying notion is that the day-to-day behavior of people in home, work and leisure situations importantly depends on the ability of a person to assess the correctness of their own responses as well as on the knowledge and skill which they possess. Thus, learning might better be considered a change in a person from being unsure-and-wrong to being sure-and-correct; it is insufficient to consider the learning objective simply to make correct responses.

INTRODUCTION

My explicit interest in trying to solve some problems of teaching was precipitated by my own experience when I first started teaching in 1962. For eleven years prior to that I had been employed as a Research Psychologist by the U.S. Air Force at their Aerospace Medical Research Laboratories at Wright-Patterson Air Force Base in Ohio, where I conducted research and worked as a consultant to design engineers and manufacturers of aerospace systems in the area of, what may be called, Engineering Psychology and also in the multidisciplinary area of Human Factors or Ergonomics (I will use the latter two terms interchangeably).

Ergonomists are essentially behavioral, social, physiological, medical and engineering scientists and professionals who develop knowledge in these areas of human science and apply it to the design, use and maintenance of technological systems and machines. The point of view assumed by most ergonomists and engineering psychologists is that machines may be considered to be extensions of a human's capacities; so that no machine, including a computer, has any reason for existence except for the use by humans (or, in some cases, by animals). Thus, computers are seen as machines which extend a person's abilities to process, manipulate, store and retrieve certain kinds of information.

During this presentation I want to

(a) provide a conceptual framework and a perspective for considering computers in teaching and testing, and

(b) describe three projects in which we have used computers to apply our basic research findings to instruction and testing.

It has been suggested by analyzers of the news that the emphasis during the television coverage of the recent national elections was on style (how the candidates looked, moved, etc.) and on the race-horse aspect (who is ahead, who is moving up, etc.) rather than on the substance and questions which would be more relevant and helpful to the voters' making good decisions. The implication is that the 'device' of television may have diverted people's interest and attention away from more important questions toward less important ones.

In a similar way, emphasizing the computer device, itself, rather than emphasizing the purposes and functions of teaching is likely to produce counterproductive attitudes in teachers, students, administrators, etc. and will direct our attention and energies to the wrong questions. Thus, I suggest that it is essential to construct first a conceptual framework for considering the numerous components which affect teaching effectiveness and costs. Without such a conceptual framework we will wander serendipitously from one component to another: From the teacher component ('The human teacher must be forced or enticed to teach better') to the student ('The student must spend more time on learning basic skills') to machines ('We must use more computers in teaching').

Not only is such component-thinking too simplistic to deal effectively with the complex teaching problems, but also such ineffective efforts are interpreted by many people as indicating that the problems are being solved when, in fact, they are not. Thus, these ineffective efforts are actually counter-productive because they divert money and resources away from more useful projects. The complexities of the problem are suggested by a recent article in the American Psychologist. Follman (1984), in a brief article entitled, "Pedagogue--Paragon and Pariah--20% of the time: Implications for Teacher Merit Pay," cites many studies from 1927 through 1977 conducted in different types of educational institutions which show "...that about 20% of teachers perceived as good, effective and best by some students are perceived as bad, ineffective, and worst by other students" (p. 1069).

As behavioral and educational scientists, the task which is most challenging and demanding of our creative talents is that of formulating insightful questions which are both important and answerable. A conceptual framework is vital to serve as a guide for us to formulate good questions concerning the use of machines, including computers, in teaching. Systems theory (e.g., Miller, 1978), which will be discussed in more detail later, provides such a framework.

I will use as a point of departure a recent article by Skinner (1984) entitled, "The shame of American education." He argues that educators and teachers have not seized the chance to improve teaching because "...the solution conflicts with deeply entrenched views of human behavior, and that these views are too strongly supported by current psychology" (p. 948). I find his comments to be very relevant as well as provocative.

My reading of his article shows that he attributes these conflicts to two general factors: (1) our failure to recognize that machines that teach are teaching machines, and are not computers and (2) a commitment of teachers and of teachers of teachers to humanistic and cognitive psychology.

What Do You Call Machines That Teach?

It is important to select an appropriate label because it

a. helps to construct the conceptual framework within which the computer is considered in teaching and testing,

b. helps to determine the attitudes and dispositions to respond in particular ways of the human teachers, and

c. affects the kinds of research and application questions which are formulated.

The label, "computer," tends to focus the attention on the machine rather than upon the purposes for which the machine is being used. Skinner (1984) has suggested that a more satisfactory and more accurate label, is "teaching machine." Skinner makes the point by proposing,

> Computers are now badly misnamed. They were designed to compute, but they are not computing when they are processing words, or displaying PacMan, or aiding instruction.... Instruction may be 'computer aided,' and all good instruction must be 'interactive,' but machines that teach are teaching machines (p. 948).

The concept that the modern teaching machine simply uses computers for the storage and control of information is shared by most learning psychologists. This may be due to the fact that the idea that machines could be used to teach was introduced in psychology in 1926, more than 50 years ago, by E.L. Pressey of Ohio State University. He developed a machine to automatically administer tests and "...made the perceptive observation that the machine could teach as well as test..." (Adams, 1980, p. 194).

A colleague (Dubois, 1984) suggested to me that the label, "teaching machine", also should be disqualified. In the 1960s, the unkept promises and failures to live up to the heavily advertised advantages of teaching machines produced a backlash and rejection of them. Thus, the label "teaching machine" may have associated with it some residual feelings of past failure and rejection.

Perhaps this is a good place to define teaching and training as the management of a learning situation aimed at the attainment of particular knowledge, skills, attitudes and performance. "Learning" may be defined here as a relatively permanent change in the behavioral potential of a person which accompanies practice. Sometimes the learning objectives may not be very well specified, but that is another topic. A "systems approach" (or general systems theory) helps to maintain one's focus on the purpose of teaching and education and helps to maintain an appropriate perspective for considering teaching machines and computers--or whatever we decide to label these machines.

A Systems Approach to Teaching

During the 1950s there emerged in many military, industrial and academic settings an increasing recognition of the importance of what may be called a "systems approach." One central idea of the systems approach, which started in engineering many years before, is that two or more components interact and communicate with each other in order to attain a common purpose. This kind of approach was being employed by ergonomics scientists to conceptualize the human as a component which interacts with machines (and other people) to attain a common purpose.

Over the past several decades general systems theory has undergone considerable development. The most elaborate and recent application of systems theory to the human being is probably contained in J. G. Miller's "Living Systems" (1978).

However, the behavioral, educational and social sciences, in contrast to the medical and biological sciences, have been slow in adopting the systems concepts and incorporating them as tools for advancing their scientific and professional condition. For example, only in the last several years has there been any substantive mention of general systems theory in textbooks in introductory psychology or in any development of behavioral theories (McConnell, 1983). A representation of a systems approach for teaching and some of the components involved are shown in Figure 1.

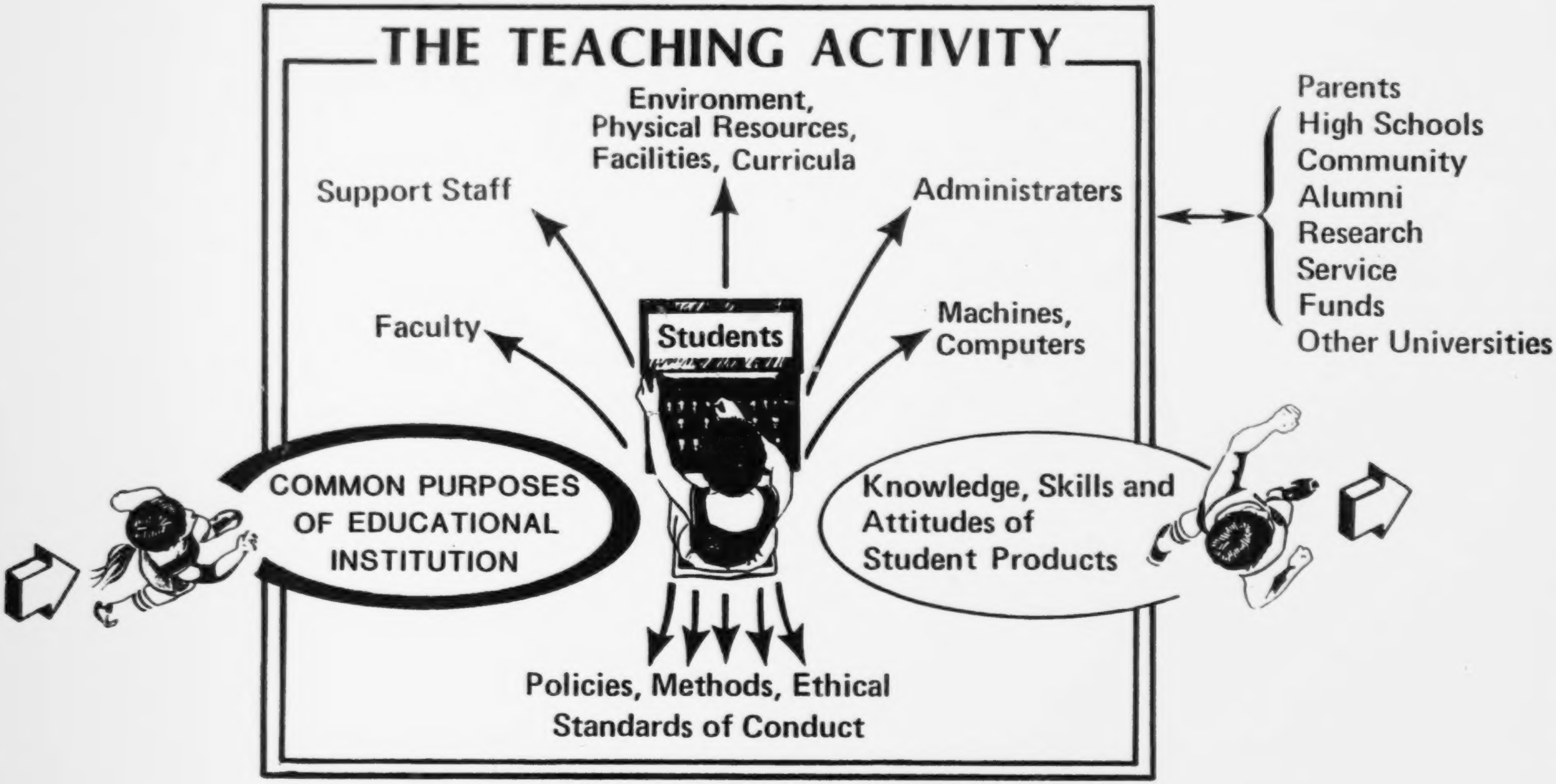

THE TEACHING ACTIVITY
Environment,
Physical Resources,
Facilities, Curricula
Support Staff
Administraters
Faculty
Students
Machines,
Computers
COMMON PURPOSES
OF EDUCATIONAL
INSTITUTION
Knowledge, Skills and
Attitudes of
Student Products
Policies, Methods, Ethical
Standards of Conduct
Parents
High Schools
Community
Alumni
Research
Service
Funds
Other Universities

Figure 1. A conceptual framework of the teaching activity in a university as a system of interacting components, all of them having a common purpose.

I have discussed systems theory in as much detail as I have because it provides a conceptual framework for considering computers in teaching, training and testing. Specifically, computers are considered to be a machine component in a system which consists of other components such as the student/learner, the textbook, the human teacher/trainer, the teaching administrators, the policies, rules and curricula, the ethical standards of behavior which prevail, the environmental factors such as temperature, humidity, noise, and the design of the learning-place equipment such as the desk, seat, informational displays, controller or marking devices and communication modes. For example, there are at least five kinds of devices (light pen, joystick, mouse, trackball and touchpanel) other than the cursor keys on a keyboard which are better for certain purposes (Weissman, 1984) to move a cursor or to mark an area on a monitor screen than are the cursor keys on a computer keyboard (the ones with arrows marked on them).

All of these components combined have a common purpose of the student learning some specified material, e.g., for the student to learn to solve specified problems, to understand some concept or to perform some specified job. In general, the purpose of such a learning and teaching system is to impart some specified knowledge or skills to one or more student learners. Or perhaps, I should say that we should design the teaching system so that it manages the learning circumstances so that the learner's performance potential is altered to be in accordance with the learning objectives.

Consider the first course which I taught in 1962--the statistics course. I delivered well-organized, substantive, interesting lectures for the purpose of the students learning the material from these lectures, the textbook, a workbook, each other, etc. To my great disappointment, the students' performance on the tests were very poor.

An analysis of this teaching problem as a closed loop system led me to the conclusion that one problem was that my interpretation of my observations of the students' behavior, their facial expressions, etc. was inaccurate. Often I believed that they were understanding the lecture material when, in fact, they were not. So an improvement in the accuracy (or validity) of the feedback information to me, the lecturer, should improve the effectiveness of the instruction.

STUDENT SELF-ASSESSMENT:
ASSESSING THE CORRECTNESS OF ONE'S OWN RESPONSES

Each classroom student was given a hand-held button, which he/she was told to press when they were not understanding the lecture. These buttons were connected by wires to a meter mounted on my lecture podium which displayed the percentage of students who were pressing their buttons at any moment in time. When the needle on the meter reached too high of a percentage, then I would stop and try to remedy the apparent misunderstanding. It should be noted that the students had to make an assessment of whether they understood or did not understand the lecture in order for them to decide whether to press or not to press the button.

A consideration of the ability of people to make these kinds of self-assessments led me into a rather extensive program of research and thought concerning the human processes which might underlie this activity and to consider the practical implications of these self-assessment responses. For one thing, my research indicates that simply requiring a learner to indicate "how sure" he/she is of the correctness of each response which is made may speed up learning (Hunt, 1982). More generally and more importantly, the research suggests that the behavior of a person in day-to-day life in homes, work, professions, sciences, sports and leisure, depends on both

a. the knowledge, skills (and attitudes) which a person has acquired from learning and experience, and

b. the ability of a person to assess accurately whether particular knowledge and skills are possessed and whether responses are correct.

Considerable research in the behavioral and educational sciences for a hundred years or so has been directed toward the identification of factors which may help to improve the ability of our educational and training institutions to impart knowledge and skills. However, little systematic or experimental attention has been devoted to what may be called "human self-assessment responding" which is the ability of people to assess the correctness of responses or actions which they are anticipating making--or--which they have already made but not yet received feedback concerning their correctness. Learning might better be thought of as a change in a person's behavior from making Unsure-and-Wrong responses to Sure-and-Correct responses. It is not sufficient, I think, to consider the learning objective simply to make correct responses.

The study of the human self-assessment responding has led me to use machines not only to conduct research, but also to apply some of my research findings. Grants from the U.S. Army Research Institute for the Behavioral and Social Sciences over the past 7-8 years have allowed me to conduct basic research (Hunt, 1984) and to develop many projects, three of which I will discuss today. They are

1. Self-Assessment Multiple Choice Tests

2. Self-Assessment Computer Aided Testing (SACAT)

3. Self-Assessment Computer Aided Review and Instruction (SACAR)

Self-Assessment Multiple Choice Test

This is a pencil-and-paper multiple choice test (see Appendix A) which requires the test taker to mark an answer and also mark "how sure" he is of the correctness of the answer. These answer sheets can be taken to the counter in the NMSU Computer Center where the Answer Sheets are read using the Optic Scanner and scored, by a computer program installed at the Computer Center.

The tests are scored to provide both the conventional percentage correct and a "Self-Assessment Score." The number of points gained or lost on an item depends on whether the answer is correct and, logarithmically, on the level of sureness indicated by the students, as shown below.

	Not Sure	x	x	x	Sure
Correct	+1.0	+2.7	+3.7	+4.5	+5.0
Wrong	+0.5	-0.4	-1.6	-3.2	-6.0

Print-outs (which can be used for the purposes shown) are provided to the instructor as follows:

a. Student Listing giving only Social Security Numbers with no names (For posting to provide feedback information to the students)

b. Frequency Distributions for Percentage Correct and for SA Scores (For posting to help students to interpret their scores and letter grades)

c. Instructor Listing giving students' names as well as Social Security Numbers

d. Item-by-Item Information giving the percentage of students who were Sure-and-Correct, the percentage of students who were Sure-but-Wrong, etc. (To help the instructor detect misinformation, evaluate the teaching effectiveness, identify instructional problems, etc. For example, if a particular test item has an unusually high percentage of Sure-but-Wrong answers then the students may be misinformed rather than merely uninformed; and the teaching methods appropriate for remedying misinformed students may be different than the typical methods designed for teaching the uninformed.)

e. Student Answer Listing of all answers given by all students (To help resolve any questions concerning what answers and self-assessments students gave on the test).

I have employed the knowledge in ergonomics/human factors and engineering psychology in designing the print-outs for the students and instructor.

Self-Assessment Computer Aided Testing (SACAT)

We are just beginning to work on two approaches using self-assessment in a testing situation. One approach is simply computerizing the paper-and-pencil multiple choice test so that students can take the test directly on a computer which (a) provides more rapid feedback to the students concerning their test scores, (b) allows a greater flexibility for the students as to the time when they take the test and (c) provides for the possibility of incorporating the self-assessment methods into other kinds of computerized testing, e.g., randomly selecting from populations of questions.

The second kind of Self-Assessment Computer Aided Testing method is focused on two aims which are (a) using the self-assessments, along with the correctness of the answer, to detect and identify specific test performance deficiencies and (b) taking advantage of the testing session to serve also as a learning session. The general notion is to consider a person as a "system" consisting of a number of components and processes (Miller, 1978) which interact to produce the individual's performance. The testing strategy is to begin questions, within some topic, at a general level and present questions with increasing specificity aimed at identifying the specific component or process which is responsible for the deficiencies relative to the broad topic or job being tested. The approach is to develop an array of different kinds of questions from which one or more would be presented contingent upon both the self-assessment response and the correctness of the answer. The kinds of questions represent different levels or hierarchies of knowledge of skills which, all combined, are required to perform a specified task or job. Simultaneously an attempt is being made to provide a learning situation to remedy the identified deficiency.

A Self-Assessment Computer Aided Review (and instruction) Program (SACAR)

This approach to teaching machines incorporates my belief (based upon my own and others research findings and thought) that the behavior of people in everyday life importantly depends upon their self-assessments. The human self-assessment process has been not only the focus of my research, but also the focus of my attempts to use computers in teaching training. Because the usual computer keyboard is not suitable for use as a teaching machine by most people, I have designed a student response keyboard specifically for SACAR (See Figure 2), incorporating ergonomics principles of design for the human user.

The keyboard provides for the student to enter their identification number, choose from a menu the specific material which they wish to review or learn, select an answer to each item when it is presented and indicate on a 5-point scale "how sure" they are that the selected answer is correct. The leftmost button is labelled "Not Sure" and the rightmost button is labelled "Sure".

Immediately after an answer to a question and an SA response is given, four items of information are presented on the student's visual display:

a. the answer response which the student gave

b. the correct answer

c. the number of points gained or lost on that item

d. the cumulated number of points so far.

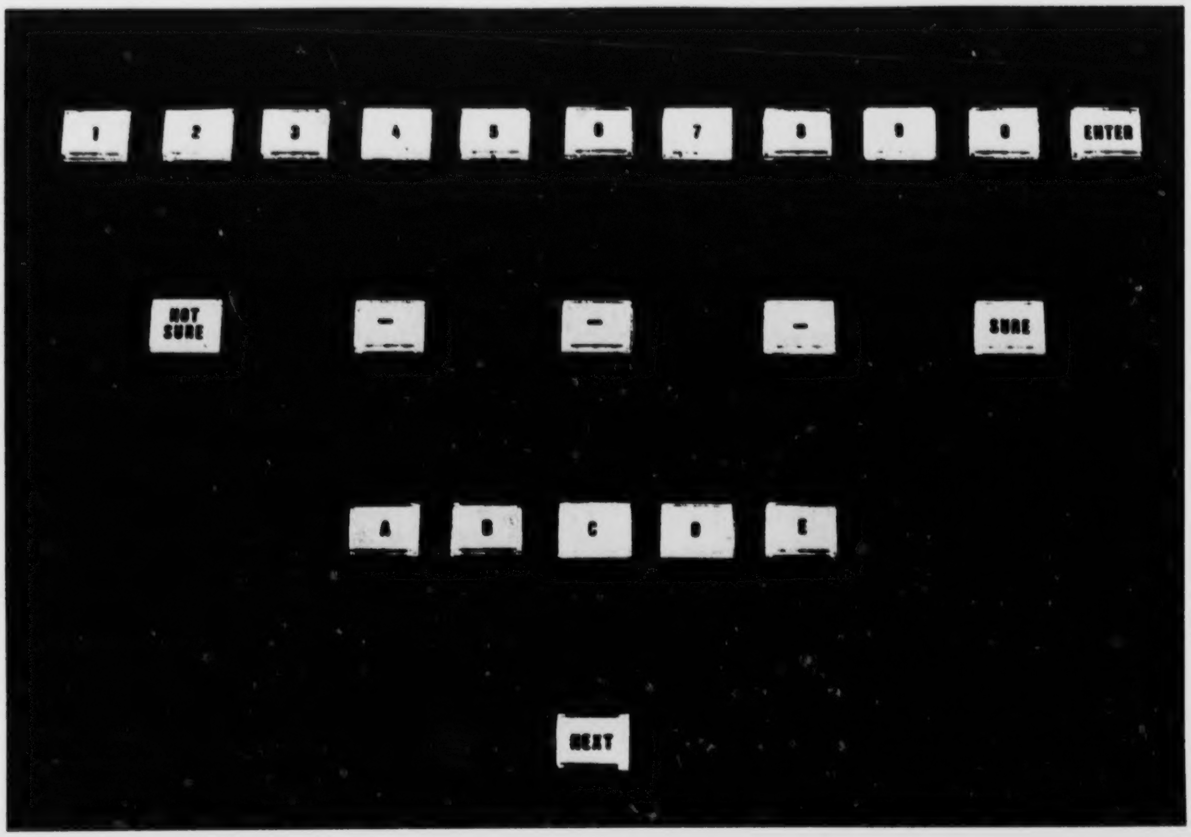


Figure 2. An ergonomically designed keyboard for human learners using the Self-Assessment Computer Aided Review - Instruction (SACAR) machine.

The number of points gained or lost on an item depend on both the correctness of the answer and the accuracy of the self-assessment. The scoring rule for assigning points is the same as is used in the Self-Assessment Multiple Choice Test.

When the student has completed the review of the chosen material, the student is provided information on the visual display concerning (a) the percentage of correct responses to provide feedback to the student concerning the correctness of the answers and (b) an SA score which can range between 0 and 100 to provide feedback to the student concerning the accuracy of their self-assessment responses. This "self-assessment" method could be adopted to almost any computer aided instructional program.

Computer Simulation of Human Self-Assessment Responding

We are also using the computer to construct a simulation of a theoretical model of the human self-assessment process which we have developed. The simulation is called SAM HUMPAT which stands for "self-assessment model of human performance and training." We are committed to what I call a Keplerian approach which emphasizes the incorporation of all that we know about the human perceptual, learning, psychomotor and other

BEST COPY AVAILABLE

psychological processes into the simulation. We are not interested in simply constructing a mimic of human self-assessment responding. I call it the Keplerian approach in honor of the inspirational work of Johannes Kepler (e.g., Dreyer, 1953) concerning his laws of planetary motion and it is in contrast to the earlier epicycle models which provide good predictions, but which are based on erroneous concepts of how planets move.

REFERENCES

Adams, J. A. (1980). *Learning and memory*. Homewood, Ill.: The Dorsey Press.

Dreyer, J. L. E. (1953). *A history of astronomy from Thales to Kepler* (*Second Edition*). New York: Dover Publications, Inc.

Dubois, B. L. (1984). Personal communication.

Follman, J. (1984). Pedagogue-paragon and pariah-20% of the time: Implications for teacher merit pay. *American Psychologist*, *39*, 1069-1070.

Hunt, D. P. (1982). Effects of human self-assessment responding on learning. *Journal of Applied Psychology*, *67*, 75-82.

Hunt, D. P. (1984). *Human self-assessment and its implications for human training and performance*. Summary Report for the U.S. Army Research Institute for the Behavioral and Social Sciences.

McConnell, J. V. (1983). *Understanding human behavior*: *An introduction to psychology* (*4th Edition*). New York: Holt, Rinehart and Winston.

Miller, J. G. (1978). *Living systems*. New York: McGraw-Hill, Inc.

Skinner, B. F. (1984). The shame of American education. *American Psychologist*, *39*, 947-954.

Weissman, J. (1984). Marking devices for computer-based training. *Training Technology Journal*, *1*, 48-54.

APPENDIX A

NCS Trans-Optic M801-17409-321

MULTIPLE CHOICE SELF-ASSESSMENT ANSWER SHEET

©COPYRIGHT 1983 DARWIN HUNT – ALL RIGHTS RESERVED

DIRECTIONS:

1. FOR EACH QUESTION, PLEASE SELECT THE BEST ANSWER AMONG THE GIVEN ALTERNATIVES. USING BLACK LEAD PENCIL ONLY, #2½ OR SOFTER, COMPLETELY BLACKEN IN THE CIRCLE CORRESPONDING TO YOUR SELECTED ANSWER. ONLY ONE ANSWER PER QUESTION ALLOWED.
2. IMMEDIATELY AFTER YOU HAVE MARKED YOUR ANSWER, PLEASE INDICATE HOW SURE YOU ARE THAT YOUR ANSWER IS CORRECT BY BLACKENING IN ONE OF THE FIVE LEVELS WHICH IS MOST REPRESENTATIVE OF YOUR CONFIDENCE IN THE CORRECTNESS OF YOUR ANSWER. ONLY ONE CONFIDENCE RESPONSE IS ALLOWED.
3. THE POINTS YOU RECEIVE ON EACH ITEM ARE BASED ON BOTH THE CORRECTNESS OF THE ANSWER AND THE ACCURACY OF YOUR SELF-ASSESSMENT. INDICATING HOW SURE YOU ARE AS ACCURATELY AS YOU CAN WILL PRODUCE THE HIGHEST SCORE.
4. DO NOT FOLD OR STAPLE THIS SHEET. MAKE NO STRAY MARKS AND ERASE CLEAN ANY MARK YOU WISH TO CHANGE.

EXAMPLE:

	ANSWER	HOW SURE ARE YOU?
	T F Ⓐ Ⓑ Ⓒ ● Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪ ① ② ● ④

YOUR ANSWER IS Ⓓ AND YOU ARE FAIRLY CERTAIN ③
THAT YOUR ANSWER IS CORRECT

WRITE IN SOCIAL SECURITY NUMBER IN SPACE PROVIDED. BLACKEN IN CIRCLE BELOW CORRESPONDING TO NUMBER ENTERED.

⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪
①	①	①	①	①	①	①	①	①
②	②	②	②	②	②	②	②	②
③	③	③	③	③	③	③	③	③
④	④	④	④	④	④	④	④	④
⑤	⑤	⑤	⑤	⑤	⑤	⑤	⑤	⑤
⑥	⑥	⑥	⑥	⑥	⑥	⑥	⑥	⑥
⑦	⑦	⑦	⑦	⑦	⑦	⑦	⑦	⑦
⑧	⑧	⑧	⑧	⑧	⑧	⑧	⑧	⑧
⑨	⑨	⑨	⑨	⑨	⑨	⑨	⑨	⑨

DO NOT MARK IN THIS SPACE – FOR COMPUTER CENTER USE ONLY ● ●

NAME ______________________________ DATE ______________

COURSE TITLE ______________________________ INSTRUCTOR ______________________________

	ANSWER	ALMOST GUESS	PROBABLE GUESS	NEUTRAL	FAIRLY CERTAIN	ALMOST CERTAIN		ANSWER	ALMOST GUESS	PROBABLE GUESS	NEUTRAL	FAIRLY CERTAIN	ALMOST CERTAIN
		HOW SURE ARE YOU?							HOW SURE ARE YOU?				
1	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	16	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
2	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	17	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
3	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	18	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
4	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	19	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
5	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	20	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
6	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	21	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
7	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	22	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
8	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	23	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
9	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	24	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
10	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	25	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
11	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	26	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
12	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	27	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
13	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	28	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
14	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	29	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④
15	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④	30	T F Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ	⓪	①	②	③	④

DO NOT MARK
OR WRITE
IN THIS SPACE

	ANSWER										HOW SURE ARE YOU?				
	T	F									ALMOST GUESS	PROBABLE GUESS	NEUTRAL	FAIRLY CERTAIN	ALMOST CERTAIN
31	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
32	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
33	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
34	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
35	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
36	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
37	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
38	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
39	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
40	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
41	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
42	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
43	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
44	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
45	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
46	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
47	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
48	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
49	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
50	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
51	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
52	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
53	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
54	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
55	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
56	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
57	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
58	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
59	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④
60	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	⓪	①	②	③	④

BEST COPY AVAILABLE

A COMPUTER ON EVERY DESK;
IMPLICATIONS FOR THE EDUCATIONAL PROCESS

by

Kenneth L. Krause

William E. Richardson

Lawrence G. Jones

United States Air Force Academy

116

A COMPUTER ON EVERY DESK; IMPLICATIONS FOR THE EDUCATIONAL PROCESS

ABSTRACT

The United States Air Force Academy (USAFA) is currently implementing a project to place a microcomputer in every cadet room and put a substantial number of microcomputers in the faculty and staff areas. In addition, the creation of common use areas and laboratories of microcomputers will allow in-class or group utilization of the computers. In all, 3000 microcomputers or more will be brought to the campus to service approximately 4400 cadets and a faculty of about 570. The second phase of the project will link all of these microcomputers and the existing computer resources together with a local area network. We overview this project explaining scope, goals, and overall plan.

The technology which has made this project feasible certainly has some profound educational implications. The general educational issues we discuss are categorized by their impact on students, curriculum, faculty, and support services. Next we present some technical (non-educational) implications. This discussion includes planning, management, hardware and software issues. Although we classify these issues as technical, they have a significant impact on the academic environments and, hence, are just as critical to the university as the educational issues.

Although the project is far from complete, we have learned a number of significant lessons about the full scale introduction of this technology into the academic arena. To conclude, we present our approach to some of the problems and review some of the lessons learned.

1. BACKGROUND

1.1 Introduction

The United States Air Force Academy completed a study of computer support requirements in June 1983. The study concluded that we needed a major expansion of the existing capabilities. The planned expansion includes at least one microcomputer in every cadet dormitory room and a substantial number of microcomputers in the faculty and staff areas. All of these small computers will be linked with the existing large scale computers by a local area computer network. In all, this extremely ambitious project will involve over 3000 microcomputers to serve the approximately 4400 students and 570 faculty.

This much technology will have a major impact on any institution. In this paper we explore some of the issues and share some of our experiences and speculations about the effects of this technology. We begin by discussing the project scope and goals. Given this background, we examine issues that pertain primarily to the education process itself including possible impacts on the students, curriculum and faculty. We next briefly explore some of the more technical issues including the managerial considerations as well as the computer hardware (machines) and software (programs) that support the project. We continue by explaining some of our responses to the problems and conclude with some lessons learned.

1.2 Project Overview

The set of initiatives to meet the future computing needs of the Academy is known collectively as the Microcomputers in the Dorms Project (MID). The MID has three broad goals:

To improve the quality of the USAF Academy graduate,
To enhance cadet educational experiences,
To provide a modern academic environment.

As mentioned, to meet these goals we plan to have microcomputers installed in every dorm room and in many faculty and staff offices and link these facilities and mainframe computers with a network. We will also install computers in laboratories, classrooms and other common use areas. Another important part of the project is to provide a useful environment of software tools for using the computers effectively.

The MID has two distinctly identifiable portions, the microcomputers themselves and the local area network (LAN). While we will deal with both issues in this paper, we view the microcomputers as the quantum leap from where we are now and the network as a lesser multiplier. The big change will be due to the enormously increased availability of computing power and the dispersal of physical control. We feel the network's impact will be one of increased convenience of communication.

For the microcomputers themselves, the envisioned uses may be classified as word processing, educational support, software packages, and program development. The most frequently envisioned use of the computers is word processing. A stand-alone microcomputer has enormous utility for preparing papers, reports and letters. Microcomputers have utility for educational support such as computer assisted instruction (CAI), computer testing, and computer managed instruction (CMI). The CAI use may be somewhat limited to drill and programmed instruction. The more interesting aspects of CAI may require equipment that is too expensive for large scale purchase. The LAN will greatly expand the CAI capabilities.

Word processing and educational programs are special cases of software packages. These prewritten packages support special applications such as spread sheets, inventory, budget or general data base creation. Finally, the more sophisticated users will need support for developing their own programs. This requires a host of software development programs such as editors and high level language compilers. All of these uses are anticipated under the MID.

The local area network has three main features envisioned with elaborations of these features as budget permits. The three categories of service are the ability to access one computer system from another, the ability to transfer files of information between systems, and audio/video channel capability. Remote access allows you to use the micro as a terminal to another larger specialized system. This opens up all applications that are traditionally available only by going to a common central terminal room such as running big programs or packages that are too large for a micro. After you have remote access and file transfer capabilities you then open up a wealth of possible applications, e.g., electronic mail, electronic bulletin boards, centralized storage of information, course material distribution (assignments, lesson plans, special handouts), electronic appointment calendars, etc. The video channels will allow in-room viewing of special lectures, required movie viewing, and ability to view lectures missed due to illness to name a few possibilities.

The project is currently in the stages of specifying the LAN and experimentation with several types of microcomputers. The micros will be specified later this year with initial delivery of half of the final number in June 1986. The other half will be delivered along with the complete network by August 1987. This timeline is dictated by budgetary constraints rather than by anything else in particular.

Since we are just in the early stages of the project, some of the ideas we discuss are speculative. Given the number of problems we have already encountered we suspect that if anything we are understating the difficulties that we will run up against.

2. EDUCATIONAL ISSUES 2.1 Curriculum

There is no question that the proliferation of personal computers in the university environment as well as the elementary and secondary schools will have an ever widening impact on our curriculum. There is no agreement on whether that impact will be positive or negative [CHOR84, OSGO84]. Two of our major concerns as members of a Computer Science Department are: 1) computers driving the curriculum instead of the curriculum being enhanced by the presence of computers, 2) the time demand on students with the increasing requirement for student produced computer products across the curriculum. We are also concerned about the proliferation of bad computer use habits (especially programming techniques) as those without formal computer science education teach the novice users. We also see tremendous potential for enhanced higher education through the proper application of computer power within the curriculum. So, how do we answer the question of where, when, and how to apply computer technology within the curriculum? We certainly don't have all the answers (any of the answers?) but can share our thoughts and planning so far.

We believe early, formal training on the use of the personal computer is important. We have a required computer science course now given to 70% freshman and 30% sophomores. That will probably change to an all freshman course when we pick up the charter of teaching proper use and programming (in Pascal) of the cadet micros. In addition, we are looking at having a required basic skills course given to all students before they start their first semester. This would include basic care and use of the machine, the institutional text editor, and any other "common" software packages. We now have a mandatory typing proficiency test here at the Academy and it is conceivable that testing on a microcomputer would largely replace this. The introductory computer science course teaches Pascal programming skills, structured problem solving, and computer literacy. We are now teaching that course on the Zenith 100 microcomputer with Turbo Pascal. The course is jammed full of information. A curriculum study underway now may expand the material to two required courses.

One advantage of a computer on every desk to the curriculum here at the Academy (largely an Engineering school) is the ability to solve and hopefully understand more complex problems because of the availability of so much computing power. E. W. Dijkstra, noted European computer scientist, has often said that the ability to solve "toy" problems, scaled problems, does not help in solving large (complex) problems. The availability of computer power should allow for the student solution of larger, more complex problems.

There is a large potential gain in the area of computer managed instruction. The availability of electronic products (lesson plans, course handouts, syllabi, instructor schedules, administrative products, computer generated tests, etc.) via the

LAN will benefit their quality, timeliness, and stability. The grading information should be made available to students more quickly to let them evaluate more realistically where they stand in a course. The state of the products that support the curriculum should improve. Achieving this may require far more administrative control than is currently typical.

Computer assisted instruction (CAI) is certainly a hotly debated issue today [CHOR84]. At the Academy, Chemistry, Physics, and foreign languages are currently using CAI. Many other departments are now looking at the possibility of CAI use when we have a micro in every dorm room. The video channels on the local area network have definite potential to support the CAI concepts. Current technology limits this somewhat so that we have not moved much beyond "page turning" at this time. One main concern is the time involved in producing CAI (about 300 hours per hour of instruction). Current CAI authoring packages help some but don't make much of a dent. We have neither the manpower nor the money to produce substantial amounts of CAI. If CAI is to prove cost effective, it may only be for very large, relatively stable courses. We will talk further about this problem under the faculty discussion.

Along with the benefits, we see some potential problems. Even though we have presented the case that engineering students may be able to solve larger problems with all the computer power available it is also possible they will fall into a "plug and chug" mentality with no understanding of the algorithms and concepts behind the computer products. This is an expansion of the same fears we saw with the introduction of programmable calculators. We see a continued inability of students to spell if a word processing "speller" is routinely used. Cheating will be easier to do and harder to detect. Even at our institution with a very strict Honor Code we see the proliferation of the problem. Electronic loss of information will present a new set of problems. We will move from the "dog ate my homework" era to the "computer ate my disk" era. The largest problem to the computer science community is the difficulty in developing and enforcing good programming and use methodology.

2.2 Students

Students will need to be more highly skilled than they are now or at least have a different set of skills. These skills will include typing, detailed level of reasoning, and, of course, computer literacy and use skills. We have found the incoming skills of freshman to be increasingly more diverse. The notion of placement exams, homogeneous sectioning, and validation become more critical. This same problem from computer science will now spill over into the rest of the curriculum with computer skills being necessary in all disciplines. We also see the effect of those students who are poorly trained in computer usage at the high school level which will again affect the whole spectrum of the curriculum. Most students will hold on for dear life to whatever software and hardware products they were originally trained on.

We now see the "terrified" or "non-detail" student in our introductory course. They have (or develop) an irrational fear (hate) of the computer. How will this student fare in a computerized curriculum? We see in this case the computer being a barrier to learning, a frustration, and a time sink. We are not necessarily talking about a "dumber" student, but rather one whose aptitudes lie in a different area that does not involve detailed manipulation.

Word processing should enhance the student's written work and the efficiency in producing that work. We see qualitative improvements in writing as a possibility with appropriate use of easily modifiable outlines and drafts. We have a fear in this area as well as in the other areas of computer usage within the curriculum that the increased expectations of the faculty will exceed the enhancement of capability provided to the student.

Much work is still to be done on the human interface for the student. What are the effects of using ill conceived software packages over a period of time? What is the frustration level when the computer response to an erroneous input is to terminate the session? How do we guarantee good human interfaces across an institution? We feel there must be a standardization of the software interfaces with the students. This requires manpower and money. There will be an information explosion with an addition to the computer power on a campus (especially if you include a local area network as we are planning). The computer allows easy access to this information which contributes to the educational process. Will the explosion be so great and uncontrolled that this advantage goes away?

2.3 Faculty

There will be and is a wide spectrum of acceptance and utilization of the additional computing capabilities among the faculty. At one end we see dictates to "use the computer" without much thought of curriculum impact and on the other extreme we see no desire or knowledge to incorporate the computer. The administration must face up to "selling" good usage on one end and some usage on the other to bring the extremes toward the middle. The ownership issue is very important. The administration must "bring" everybody along starting with the planning phase.

Utilization of faculty time is a key issue. We have already mentioned the high cost of creating CAI (about 300 hours to 1). Will there be "credit" given for creating software for student use? How does the time count toward the "publish-or-perish" game points? The answers can certainly have an impact on faculty size and organization. The computer support environment is going to have to take on a new size and probably a higher position within the organizational structure. Provost for Computing is a new position springing up at many universities. There must be a large effort in the creation and management of any large scale computer project, especially one with the educational implications such as a computer for every student.

Part of that planning must include the training and education of the faculty. This process should not be left to osmosis. The faculty must understand the implied workload when they levy computer requirements on a student. Standards of usage both in consistency and practices must be drilled into the faculty and then enforced. Without this education and enforcement there will be a proliferation of poor practices, especially by the computer semi-literate. Software standards must be created and maintained in such a way to make them enforceable. The computer science department and computer center need to play a key role in this area and the administration must provide the proper mixture of carrots and sticks to cause the necessary standards and practices to happen.

Faculty "survival," like the students', will become more tied to the willingness to incorporate computer technology into the curriculum. What about the "cyber-phobic"? Will fear of computing curtail a faculty member's career on the computerized campus? The hope is that productivity will increase for faculty members but there is concern for the casualties.

2.4 Support Services

There is a whole spectrum of new services that a university must provide if they are to be successful in putting a computer on every desk and having a positive impact on the curriculum, students, and faculty. A big part of these services will include the training for faculty and students. As said before, it is important that this not be left to self-instruction. Bad computer use habits can last a lifetime.

There should be a well coordinated and advertised software library function that has the proper standards to allow software sharing. In addition there must be both software and hardware configuration control to maximize system usage and minimize confusion between departments and different student classes. If a local area network is installed (as we are proposing at USAFA) then there is a large, additional file and mass store management problem.

Security will be a prevalent issue (especially with a local area network). Academic cheating is an unsolved problem with all the capability for masking that cheating with the computer tools available. The privacy of one's own dorm room will add to the cheating problem. If other university services are available on the same network that the students use, the security problem becomes one of nightmare proportion.

If a network is part of the services provided along with a computer on every desk then the network services must be provided. Electronic mail, electronic bulletin boards, electronic filing cabinets, node to node communication, electronic calendar and appointment scheduling, acknowledgement announcements, and large file transfer and storage are some of the proposed user services here at the Academy. We are having a very difficult time sizing the network server computer systems because users cannot project how much and what kind of service they want from a system they have never experienced. We sent an extensive survey to the Academy community. That survey is discussed later in this paper.

3. TECHNICAL ISSUES

To bring a project of this type to fruition without greatly perturbing the academic environment requires significant technical expertise as well as an in depth understanding of the educational process. It should be understood that the project management issues that we will discuss here are just the first step in a three step approach to the technical life cycle. The other two steps, system development and operations, are vitally significant but beyond the scope of this paper. Project management, as we now see it, consists of organizational, management, and planning issues as well as technical issues relating to the hardware and software.

3.1 Organization

It quickly became obvious that the standard laissez-faire approach to changes in our university environment would certainly not suffice for putting a microcomputer in every dormitory room, much less for a local area network to link them together. Every organization which might have by default taken the lead in this type of project (e.g., the computer center, the Computer Science Department, the computer based education working group, etc.) was a bad choice to lead such a project. Generally, these organizations lack one of three credentials: expertise in the area of computer science, expertise in the area of education, or a non-parochial insight into the goals of the university as a whole. Therefore, it was determined very early that a separate organization would be required if the technical environment across the Academy was to be affected. Due to the last of the three credentials noted above, the level of this organization had to be very high in the administration of the university.

Initially, we set up a high level working group to study the issues involved in the proposed project and make some recommendations for milestones and plan of attack. This working group was chaired by the Vice Dean with a number of special interests represented. This group was strictly a coordinating body which levied the actual research and work to other agencies. It was not too long until the working group had outlived its usefulness -- there was an obvious and overwhelming need to have a Special Project Office (SPO) as a single technical and managerial point of contact. The head of this SPO had to be someone who filled all of the criteria noted above. He also had to have sufficient resources available to him to control the timeliness and quality of the decisions he would make about the project. Typically, it appears that the head of such a SPO should be an insider who can devote his/her full energy to the project. This of course is costly because an individual with those credentials will not be sitting idly by but instead will be fully employed in some other important capacity. From this experience it has become clear why some universities have taken the costly step of appointing a Provost for Computing.

3.2 Planning

The reason the project organization is so important to the automation of a campus is because the planning such an organization does affects the academic climate of the university for many years into the future. There are a number of planning issues which a project of this magnitude brings with it. However, the one significant point which must, at least today, be kept in mind while planning for automated campus is that no one has ever done a project of this magnitude before -- there is no reservoir of cohesive experience from which to draw. Therefore, the experiences must be synthesized from fragmented or scaled down experiences in many unrelated environments and fields. The potential cost of bad planning, therefore, is likely to be very high. The following points illustrate some of the aspects of planning that cannot be neglected to include computer technology as a full partner in the curriculum.

The foremost planning consideration is to develop predefined objectives and goals for the campus automation project. It is sometimes alluring to join the move toward the latest and greatest without the difficulty of considering what that means to the curriculum and the university's other goals. We at the Academy almost fell into that trap. It should be noted that it is not a foregone conclusion as some would have us believe that the automation of the campus is appropriate to all institutional objectives.

Certainly, some form of cost benefit analysis is called for. We are estimating that the non-software cost of our project will be in the area of $20 million. This must be combined with the non-tangible costs to determine the actual cost of the project. Additionally, the risk analysis of this sort of project must be considered. Since a project of this magnitude has not been done before and much of the software and hardware is not currently available off the shelf, the risks are higher than for almost any other large scale projects that would be considered in the academic environment. Consequently, definition of the project objectives allows the potential gains to be weighed against the likely costs and risks. The other advantage of having defined objectives rather than allowing the much simpler ad hoc approach is that all other plans, changes, requirement priorities, progress, etc. can be justified or measured by comparison to these goals.

The second major portion of the project planning is requirements definition. There are several important points to remember in the definition of technology requirements in the academic world. The first point is that the user is "clueless" to what he actually wants or needs. (Throughout this, when we speak of the "user" we are speaking of the departments or courses which will require student usage of the technology.) This translates into an inability to remotely, let alone precisely, define the requirements to meet the objectives and goals that have been set forth.

So what is the answer? We did a number of things which are probably a minimum set for determining the requirements accurately. The first step is to bring all of the potential users to a minimum level of understanding of the technology, its potential in academics, and its limitations. The second step is to then ask the users to help brainstorm the possible uses for the technology. We did this in the form of a survey. If a survey is used, it must define the ground rules for the user as far as objectives and goals that have been determined for the project. The survey itself needs to allow the respondent to "crystal ball" his usages or it will be simply a reiteration of current systems. The other side of that, however, is that the survey cannot be totally unstructured or the results will be impossible to interpret and use to define the requirements. Finally the survey must provide some estimates (no matter how shaky) as to the amount of usage by activity that is predicted by the user. (There is certainly another advantage of asking the user what he wants -- so that eventually he can take part of the blame!) The third step in determining requirements, really has two parts: reducing the usage data from the users and determining what other peoples' experiences have been in projects with similar goals and using similar technology. It is at this point that you realize that not only are the users still clueless but they also know just enough to be dangerous. The usage data will require significant massaging by professionals to make any sense at all, and even worse, the data from the different users will be contradictory. The next step, requirements definition from the usage data, will undoubtedly end up as a compromise which will not perfectly suit any one user's needs. Finally you must crank in a factor for the requirements of the future since, as previously mentioned, this is likely to be a long term investment for the university.

The second important point to remember in requirements planning is that you are creating a monster. Once the process noted above has been started, you begin to educate the users to all of the possibilities of the technology. That is, be prepared for the requirements to change as often as you let them. This is of course, not simply a problem of clueless users getting a clue, it also is caused by the political environment of the university, funding, changes in technology, the phases of the moon, etc. It is likely that even the hard and fast objectives of the project will go through some similar modifications.

The third point to mention in the planning for requirements definition was alluded to earlier. If the user is not part of the process, he will typically not be supportive of the project and hence may well cause it to fail. In a high technology project, especially in a typically conservative environment, the element of highest risk is often not the technology itself but the acceptance of the technology by the users. The only way to solve this potential problem is to give the users a sense of ownership by making them perceive that they are involved in the planning.

3.3 Management

Although we will not discuss the later stages of the project life cycle (system development and operations) there is rationale for looking ahead to those stages long enough to highlight the resource management problems we will see. The most striking problem involved in management of widely dispersed microcomputers and a local area network will be the decentralized nature of the technology. In most institutions the control and management of computer resources has traditionally been centralized or at least centralized within departments. This traditional approach to technology management is likely to be a thing of the past. Maintenance, resource accounting, expendable support, security, configuration control (both hardware and software), and operator support are examples of resource management problems which cannot be adequately handled by the traditional approach when microcomputers are spread through out the institution.

3.4 Hardware

A number of issues arise when considering the hardware that will be required to meet the stated requirements and satisfy the institutional goals. As previously noted, one of the first hardware problems will be to specify the systems sufficiently to insure acquisition of the proper equipment. Since this specification is based on the compromised requirements definition there is significant room for inadequate or imprecise system specification. Vendors the world over will always find the lowest common denominator in your specifications and only give you exactly the capability you specified and nothing more. When considering leading edge technology like local area networks, system specification is complicated by lack of experience with the systems, inexact standards and jargon, and a muddy definition of the line between what is available off the shelf and that which requires very expensive R and D efforts. Some schools, like Stanford, have made their microcomputer specifications so loosely defined that a very large number of systems will satisfy them. [OSGO84] We have determined that the cost of this type of flexibility in terms of training of multiple systems, maintenance, commonality of labs, etc. is prohibitive. Therefore, our specifications must truly represent our total requirements.

3.5 Software

There is little doubt that software is the most underrated problem in the adoption of the current technology. While hardware has decreased in price by orders of magnitude in the past 20 years, the complexity and cost of software has increased more rapidly. It is now all too easy to determine hardware requirements and develop the appropriate specifications without even considering how the requirements and specifications will affect the software. Computer scientists have now come to the conclusion the exact opposite approach should be taken: the software requirements and specifications should be developed first and then the hardware which best adapts to this software should be selected. We cannot attempt to solve the software problem in this paper but we will list a few of the most difficult software dilemmas.

The high cost of software is a result of many factors. Software development is very labor intensive and currently cannot be effectively automated. Once a program is written, it is virtually never shared or reused; therefore, changes in goals require substantial reinvestment in new software. Software almost never works correctly, and software maintenance is even more costly than software development. The cost to a faculty member in time and energy is ever more profound because of the educational issues that must be wrestled simultaneously with the technical software development issues.

There are, however, some approaches to software which can help to control the costs somewhat. Standards of code development and documentation for in-house software are, perhaps, the first techniques to try. These standards make it possible to maintain and use software packages even after the author has departed the institution. We have seen innumerable examples where perfectly good software had to be scrapped because the only person who understood it was no longer available to keep it running. Some form of sharing and coordination of software is important to keep from reinventing the wheel when a new package is needed within a department. This means a single agency must "store" the software packages and control the configuration of these packages (that is, changes to packages must be made to all users of the same package in order to not reinvent the wheel on maintenance either.) Information about purchase of off the shelf software should also be available from a single reference librarian since many of the departments will be looking for commercial software for similar applications. Finally, education of the administration and staff about software and the potential difficulties is required to hold the software costs in check.

After cost, the second most important software dilemma is frustration. In an educational setting, frustration can represent a significant deterrent to student achievement. Three separate software problems have brought frustration to our students: software which does not work or is not user friendly, software examples which do not follow the standards we require of them, and contradictory user interfaces on software packages. The first problem is obvious, if it does not work or is difficult to use then the student will not bother with it. If we present to the students a two faced approach to software (e.g., you must follow the rules of software creation but I am above the rules), they will quickly question the validity of the rules and also choose to ignore them. Finally, consider the student who has four different courses requiring text edited papers. If all four courses specify different text editors with different embedded commands and different keyboard configurations, the student will probably not do well in any of the courses due to unnecessary confusion they have caused him. Standardizing user interfaces on all packages used within the institution and standardizing the most common software packages (like editors and spreadsheets) among all courses will do much toward solving this last problem.

4.0 THE USAFA APPROACH

One of our most significant steps in the right direction was the recognition of the need for high level planning and involvement. While the project involves other major organizations in the Academy, the Superintendent recognized that academic requirements were the driving factor. Thus most of the required organization is under the Dean of the Faculty. Liaison is maintained with other elements but the driving force, the MID Special Project Officer (SPO), is a faculty member. As was mentioned earlier, anyone qualified to do this job already has a full time job and this is true in our case. Our SPO is the Head of Electrical Engineering and was chosen for his experience in managing other large development projects. Supporting him is a Director of Plans and Programs and a Director for Educational Research working for the Dean. Fortunately these are full time positions. The Director of Plans and Programs is responsible for providing the "big picture" planning for the SPO. The Educational Research Director assists and concentrates on educational uses such as CAI. All these people work closely with the Director of the Educational Computer Center, who is currently responsible for the existing computing resources.

Because of this high level organization, we are able to take a better look at the full impact of the project. For instance, we have addressed some of the problem of faculty education by putting together training courses, both for low level "How to do it" instruction and executive level "What's this going to do to my department" education. We have also published standards for software development to combat the "Fred is the only one who knows how to make this work and he left!" syndrome. We require a certain level of software practices and documentation for any programs that will be configured for student use (enforcement is another matter - probably unsolvable!).

Another top level decision is that there will be a standard software environment. What this means is that there will be institution approved packages and programs for common use. This prevents a student from being forced to learn different word processors for different courses, or different spreadsheets or other packages. The student can then get on with using the computer as a tool and not have the tool get in the way.

We are also taking a complete look at the entire curriculum over this year. A possible part of the outcome may be the creation of another course in basic use and computer skills. This course would quite likely be given to freshman prior to the start of Fall classes so that they could at least use word processing capabilities for paper writing. More sophisticated use would follow in other courses.

Prototyping is an important part of any new large scale project. We used the Honors version of our Introduction to

139

Computer Science course this fall to try a microcomputer based course. Based on our success we will have all 600 students in the introductory course on the micros this spring. The Mathematics department will also use some micros for some of their courses. Several departments are experimenting with micros for word processing applications.

Finally, we have managed to get a large number of people involved in planning and watching others with similar projects. The experiences at Carnegie-Mellon, Drexel and others give us some confidence that others are proceeding along similar lines (and are encountering similar problems!).

5.0 CONCLUSIONS

All big endeavors have a list of "Things You Have Learned" and "Problems To Watch Out For." We present a list of some of ours in rough order of importance.

(1). Computer Systems Analysts have a saying that goes something like, "It is too expensive to do proper planning. It is more expensive not to." It is true. Don't fall into the trap of the zealots who want to "buy hardware now, plan later." Of course requirements analysis is boring, hard, and a consensus is difficult to achieve. However, you should ask for volunteers of those who want to immediately buy a bunch of hardware and start writing programs and then shoot them!

(2). New management philosophies may be needed. The old idea of the hardware being so expensive that we have to centralize it, while not dead, is seriously ill. Inexpensive hardware makes many of the old ways of controlling ownership obsolete. However, many people overlook the enormous cost of software development and particularly software maintenance. So to save the inexperienced from themselves, to promote sharing of software, and to cut down on student frustration due to inconsistent environments, some form of centralized software control and standardization may be needed. We also require central control for network facilities. How to go about this and maintain requirements for flexibility is unsolved as yet.

(3). Poor computer habits are very hard to break. We want the technology to move us forward rather than set back the field 20 years. Insurance companies offer discounts to teenagers who take a driver's education class rather than learn from their grandmother. Similarly, in computer science we find the students with the most difficulty in our introductory course are those who were self taught or learned bad habits in high school. Plan for proper education of students and faculty.

(4). Information security will become an enormous problem. We see two primary aspects of this, cheating and loss of work. With word processors and floppy disks it becomes a thing of relative ease to plagiarize another's work (with or without their knowledge). We don't know quite how to fully address this problem. The problem of information loss is one that can only be learned by painful experience. You can tell students to back up their work and save copies but until they suffer from it, many won't pay attention. We have seriously considered incorporating a planned "painful lesson" in our introductory course (Attention students! We have just run a magnet past all the machines. Those of you who didn't follow backup procedures will have to type that 1000 pages over!).

(5). We will close with one last shot at the topic of planning. You can't do enough, and while you are at it, think about the future. Don't paint yourself into a technological box. When you establish requirements, ask people to "blue sky." You may not be able to afford some of the grandiose things this time, but try to avoid decisions that preclude flexibility. Finally, realize that planning should be a process with feedback. Try as you will, you can't plan for all problems and potential uses. Many users won't have the foggiest idea what to do with a computer until they get one. Then all sorts of wonderful uses and horrible problems pop up. Formally incorporate this feedback into your plan. It won't always make things go smoothly, but you are miles ahead of poor planning.

REFERENCES

[BRAY 84] Bray, David W. "Using Personal Computers at the College Level", Computer 17,4 (April 1984), 36-45.

[CHOR84] Chorover, Stephan L. "Cautions on Computers in Education", Byte 9,6 (June 1984), 223-226.

[Gale83] Gale, Douglas. "Cornell's Strategy for the Microcomputer Revolution", EDUCOM Bulletin Fall/Winter 1984, 9-11.

[KING83] King, John L. "Centralized Versus Decentralized Computing: Organizational Considerations and Management Options", Computing Surveys 15,4 (Dec 1983) 319-349.

[OSGO84] Osgood, Donna. "A Computer on Every Desk", Byte 9,6 (June 1984), 162-184.

(ZIMM84) Zimmerman, Frederick. "A Z100 on Every Desk: Clarkson College", Sextant March/April 1984, 8-14.

WRITING CAI TUTORIALS WITH PILOT

by

Robert Lucking

Old Dominion University

WRITING CAI TUTORIALS WITH PILOT

No frustration among those of us in higher education is more ubiquitous than recognizing that our students do not have sufficient command of a subject to study effectively and to capitalize fully on our lectures. Most instructors devise their lectures with care, planning a scrupulously detailed sequence of topics, providing rich details and elaborate examples. However, we recognize that lectures alone do not provide the familiarity, the depth of experience with the topics, or the immersion we seek. By the very nature of the structure of experience in typical classes, students can only be given an acquaintance with the material, and many are left merely with a sense of "covering topics" or "exposure" to ideas. To supplement our instruction, we select textbooks which we hope our students will read and study as a personal challenge to master the material.

Left on their own, however, students frequently read texts unable to synthesize lecture and readings in a significant manner. They become concerned about whether they will be tested on lecture material or readings, and left with what seems to them as no other option, they engage in "psyching out the prof" in selecting the material they study. Even though we hope that the relationship between reading and lecture is clearer than this, students often feel victimized or haunted by bad luck when their guesses are incorrect. As a result, they frequently feel that they are not in control of their academic success and are subjecting themselves to hoop-jumping necessary for the degree they perceive as an entry ticket into a profession.

Many professors who have come to this realization know the powers of computer-mediated instruction, and they have read of or have seen what some people in other departments or institutions are doing with computer assisted instruction. Yet they soon learn that writing computer tutorials for students under most circumstances is more complicated than they may have imagined. First, they often find they must master a computer language such as BASIC. While many academics have developed a knowledge of word processing or even fundamental BASIC, they do not have the time to become expert programmers. They pride themselves in their level of mastery of their discipline, and they fear that their competence as scholars will be threatened if they construct simple or shoddy tutorials for their students. Secondly, use of mainframe operations frequently is taxed, and terminal time remains at such a premium that Computer Science professors look upon others' use of their equipment as meddlesome interference.

With the decrease in the cost of microcomputers, more institutions are developing microcomputer laboratories or general-purpose nests of microcomputers. Fortunately the authoring language, PILOT, is available for most microcomputers, and this language is designed solely for writing drills and tutorial materials for students. By writing these additional study aids, professors from any discipline can provide students with a means of reviewing ideas or applying concepts both from lectures and from readings.

PILOT is probably the least time-consuming language to learn for writing tutorials because it was designed solely for this purpose. Other authoring languages have been developed for microcomputers, but PILOT is one of the earliest such languages and has received considerable field testing. The commands outlined here will be those used in the Apple Computer version, but these are quite similar to those versions designed for other computers. As Figure 1 illustrates, knowledge of no more than a dozen of these commands is all that is required for the framework of most tutorials, although more colorful effects can be generated using still other commands. Compared with other computer languages, PILOT's command structure and sequence are more like English, and each of the commands is descriptive of the function it performs.

Figure 1

Fundamental PILOT Commands

```
*label
T:text
R:remark
AS:accept single character
M:match  (! is 'or') (& is 'and')
TY1:respond to first yes match
TY2:respond to second yes match
JN1:jump to label name on first miss
JN2:jump to label name on second miss
U:use subroutine name (and it will return)
   subroutine has label and has E:as ending
W:3 waits 3 seconds
TS:S2 type specify; 2 is tall
TS:T2 type specify; 2 is thick
TS:I inverse
TS:N is normal
A:accept input from the keyboard
D:N$(20) Dimension numerics
A:$N$ accept a string variable
A:#N accept a numeric variable
T:$N$_,you are nice.  Uses a string variable
   name
```

By merely typing "T," the writer can specify what text is to appear on the screen. "W" indicates that the program is to wait a prescribed number of seconds before continuing, and "M" specifies what constitutes a match when the student enters a response to a question. These commands are perhaps best understood when viewed in the context of an actual sample lesson, so an example is provided (Figure 2).

Figure 2

Sample Program

```
*LABEL
T:WELCOME TO THE TUTORIAL FOR COMPUTER ONE
T:
T:IF YOU HAVE QUESTIONS, PLEASE RAISE THEM IN
   CLASS
R:NEXT IS WAIT
W:3
R:NEXT IS A CLEARING OF THE SCREEN
G:ES
T:GLAD YOU ARE SEEKING EXTRA HELP
W:3
G:ES
D:N$(30)
T:WHAT IS YOUR NAME?
R:NEXT IS AN INPUT TO A VARIABLE.  NOTICE:THE
   SPACE WHEN THE VARIABLE IS USED.
A:$N$
T:$N$ , I'M GLAD YOU HAVE COME FOR HELP
W:2
G:ES
R:NEXT IS A COMPUTE COMMAND TO START A COUNTER.
T:IS TAXITE A MINERAL OR A COMPOUND?
A:
M:MINERAL
CY:N=N+1
TY:GREAT!
TN:SORRY, YOU ARE WRONG.
T:WHAT IS THE SYMBOL FOR TAXITE?
A:
M:C^2H^3
CY:N=N+1
TY:FANTASTIC!
TN:NO, THAT IS NOT RIGHT.
T:$N$ , YOU GOT #N CORRECT
```

This tutorial begins with a label as indicated by an asterisk; a label allows the program to return to this point from another portion later on, for example, if the student wished to repeat the lesson. All statements which are preceded by an "R" are merely remarks to help document the function of that portion of the program; these statements do not appear in the lesson as it is run. The statements preceded by a "T" constitute the text of the tutorial itself, and "W" specifies the wait period before the program advances. The Statement, "G:ES," is a graphics command which erases the screen so that the next portion of the program can be shown on the display. An "A" commands the program to accept input from the student. The "A: N$(30)" is used to reserve in memory the student's name; in the case of the sample provided, the name can be 30 characters long. Because the name is kept in memory, it can be used at any point in the lesson including the very end where the student's success ratio is reported.

The core of the program, however, is the questions themselves and the means of checking the students' answers. Questions follow a "T" as the usual textual material. An "A" accepts the student's answer, and "M" stipulates what the instructor designates as the correct match to the answer to the question. The "CY: N=N+1" command keeps a running tally of those questions answered correctly; if "yes" a match has occurred, one is added to the value of N, the number correct in this instance. The text that next appears on the screen is dependent upon the accuracy of the student answer. *The "TY" indicates what text should appear if* "yes" there has been a correct match with the answer specified in the "M." In the example provided, the program displays "Great" if a match has been found and "Sorry, you are wrong" if a match is not found. After this first question is answered, the program simply proceeds to the next question.

While this example is brief and unadorned, it demonstrates the simplicity of the language and its mnemonic qualities. Obviously more complex manipulations are possible, and a number of other commands are available for more elaborate screen designs or use of subroutines. PILOT can be used to develop "branched tutorials" which jump to varied parts of the program depending on the student's success rate or wishes. For example, if a student misses a prescribed number of questions in a row, the program can branch to a loop of remediation materials. Similarly, if a student does not understand certain terminology, the program can offer a glossary of terms and definitions. Additionally, PILOT (at least in the Apple version), includes a graphics editor and a sound editor as well, and these features can be entered into the tutorial.

Relative to other computer languages, PILOT is easier to use for the computer novice, though it retains the potential for expansion as the writer becomes more experienced. An authoring language such as PILOT offers several advantages over other computer languages. It allows entry of text as simply as typing, and the writer must memorize only a small number of commands. It can generate sound or graphics rather easily as compared with the very complex commands that are required by other languages. It is flexible in its testing for correctness of student answers and can accommodate misspellings. Additionally, with the use of something referred to as "wildcards," it can accept alternative answers or lengthy responses from students.

Especially useful also is the integrated quality of the program itself. The program is menu driven, allowing easy access to either the text editor, the sound editor, or the graphics editor. The user is prompted at critical points in editing, and the program contains its own copying function and record-keeping routine as part of its general utilities. While the hardware configurations are yet to be developed and exploited in the field of education, PILOT is designed to be interactive with video devices; as time passes, many exciting possibilities will probably be realized.

As with any computer language or software program, one must also be aware of its disadvantages. PILOT is designed only for writing tutorials, and it does not lend itself to other uses such as numerical analyses or data manipulation. Its simplicity does not lead to broad utility beyond writing tutorials or other similar lessons. Secondly, programs tend to be long and demand considerable computer memory. As computer memory standards increase, this factor is less of a concern, but the nature of

PILOT does not lead to truncated structuring or looping patterns that can be performed by other languages. Additionally, the graphics capabilities of PILOT are limited to rather rudimentary displays, and artistically refined or extremely detailed graphics images are not possible.

The language itself, however, is reasonably simple to learn and can be mastered by anyone willing to devote the time to the program and its accompanying documentation. Becoming proficient at tutorial design may represent a greater challenge. Few members of academe have had this type of training, and structuring appropriate lessons from smaller segments is more complex than may at first appear, particularly if the power of the computer is to be appropriately exploited. If lessons are not developed with care, they will be little more than electric pages of text or aimless querying, and at its worst a poorly designed tutorial might result only in confusing students.

The writer of computer tutorials must plan the structure of the lesson in great detail before entering it into the computer. The major "path" of instruction should be mapped out in flowchart form; remediation loops or enrichment options can then be added later. The core of the lesson must follow sound principles of incremental learning, assuring to the greatest degree possible the student's success at each level and mastery by the conclusion of the lesson. Writing these lessons, however, will force the instructor to be very precise in analyzing and presenting the course material.

As computer assisted instruction grows from its present state of infancy, a number of questions will have to be answered. At this point, we know very little about what types of content can best be taught or reinforced using computers. An initial impression would suggest that factual material requiring multiple rehearsals would be most suitable, but we have yet to face questions of the effects of convenience and novelty. At what point will students find a computer screen less interesting and accommodating than a textbook and a comfortable chair? Further experimentation must also be conducted on styles of content presentation as they influence learning; for example, further research is needed comparing the efficacy of deductive and inductive approaches. We need to become more aware of what is known about superordinate and subordinate structures in the field of prose processing as it may apply to computer assisted tutorials. In short, we will need to see the development of computer assisted instructional psychology which keeps pace with computer hardware development.

These questions will require considerable controlled research, but our own experimentation will also help set the direction of those efforts. At the moment, microcomputers sit awaiting use during parts of the day on many, if not most, campuses, and faculty members from all disciplines can begin exploring new strategies to supplement their instruction with tools presently on the market. PILOT is a viable and powerful language which requires only the willingness to experiment with technology which may be establishing the currents of instruction in higher education for the next several decades. Perhaps we have yet to realize the potential which already exists to involve students in our disciplines in profoundly significant ways.

COMPUTERIZED TESTING FOR BUSINESS STATISTICS: A CASE STUDY

by

Paul H. Randolph

and

W. Jay Conover

Texas Tech University

155

COMPUTERIZED TESTING FOR BUSINESS STATISTICS:

A CASE STUDY

By

Paul H. Randolph
and
W. Jay Conover
Information Systems and Quantitative Sciences
Texas Tech University
Lubbock, Texas 79409

INTRODUCTION

For some years the Department of Psychology at Texas Tech has had an experimental testing laboratory using a small scale mini-computer, a PDP 11/34 with 256K words of mos memory supporting 28 terminals. Of the 28 terminals only 22 are in the testing laboratory, the remaining being used by instructors and graduate students for research and other analysis. This laboratory was set up initially for the introductory psychology course. During the 1982-83 academic year conversations between Psychology and Business resulted in the addition of the business statistics course to the experiment. The statistics experiment began with the first summer session of 1983 and has been used every semester since that time, with 2,000 students per year. This paper is a summary of our experiences with computerized testing and some recommendations for extensions of the testing program.

THE TESTING SYSTEM

The testing procedure consists of twelve multiple choice examinations of ten questions each. Questions for tests given each individual are selected by a stratified sampling process, with each item having its own pool of questions. This insures that every test is complete, since each strata can be designed to concentrate on a small part of the material. By selecting one question randomly from each strata, every test is essentially unique; that is, if the set of questions for each strata is large enough, the probability of two identical tests being generated is extremely small.

The textbook for this course is the one by Conover and Iman [1]. The examinations do not quite follow the chapter configuration, but come close. The first six examinations are over the first six chapters, respectively. Exam 7 is on the first half of chapter 7, since hypothesis testing seems to be an extremely difficult concept for the students to grasp. Exam 8 covers the last half of chapter 7 and all of chapter 8. Exam 10 covers chapters 10 and 11. Exams 11 and 12 are on chapters 12 and 13, respectively. There is no final comprehensive examination.

For each examination the corresponding textbook material was divided into ten exhaustive strata and a set of 10 to 25 questions were constructed for each stratum. The computer has an upperbound of 25 questions and cur lowerbound is 10. Most of the examination questions were constructed from the exercises in the textbook. Generally, the textbook exercises were used verbatim, but with the numbers changed. Some exercises were modified or shortened slightly to fit the examination structure. The students are given a list of all questions used on each examination at the beginning of the semester.

The basing of examination questions on textbook exercises has several advantages and almost no disadvantages. Among the advantages are the following: 1) The student has a chance to see, read and work the exercises prior to the exam, thereby eliminating the possibility of misunderstanding or misreading the statement of the corresponding question during the examination, 2) The student has a positive motivation to understand how to do each and every exercise that is assigned, quite unlike homework that is turned-in where a large percentage is copied, 3) It is not cheating for the student to seek help in understanding textbook exercises from peers or from instructors -- in fact, this is encouraged. The obvious disadvantage to basing examination questions on the textbook exercises is that the examination is textbook bound and therefore the examination becomes obsolete when the textbook is changed. The obvious solution is to choose the best textbook on the market and stick with it.

SOME BEHAVIORAL ASPECTS

Initially, we had no due dates for the individual examinations. This worked very nicely the first time the computer testing was used, which was during the first summer session with only 200 students, many of whom took examinations ahead of schedule. During the second summer session many students procrastinated, and long lines formed at the terminals during finals week. When this same pattern of procrastination began to appear in the fall semester, we initiated a system of deadline dates for the individual examinations.

Since there are 12 examinations in the course, there is almost one exam per week in a semester. We have three make-up periods, one after exam 5, one after exam 9 and one at the end of the course. In these make up periods the student may make up an exam missed or repeat one in an attempt to raise a low score.

Initially, students were permitted to repeat each examination as often as they wished, with a different test generated at each trial. The theory was that if the student repeated an examination after a period of study, reinforcement would occur.

To avoid student gamesmanship, the grade that counts is the last one. Unfortunately, the reinforcement idea was good in theory but did not work out well in practice. Instead, many students tended to engage in test-roulette; that is, they would repeat each examination over and over without study, hoping that luck would give them a good score eventually. Some students would take a test up to 22 times!

As any student of queueing theory knows, such over-utilization of the service facility quickly produces excessively long lines, and we were no exception. During the 1983 fall semester it was not unusual for students to be in line as much as three hours. As might be expected, the lines always seemed to develop on Sunday night, the due date for all exams. As a result a few reforms in the administration of the examinations were explored for the 1984 spring semester.

First of all, a bonus and a set of penalties were established. A student who took a ten-point test only once received a bonus of one point. Then beginning with the fourth trial, the student was assessed a penalty of a point per trial. The number of repeats dropped to near zero and so did the length of lines. Furthermore, the amount of study prior to the first attempt increased dramatically. Students were quick to realize that they had to score two points higher on the second attempt in order to improve themselves and this was not easy to do.

A second reform was to change the deadline dates from Sunday night to Saturday night. This resulted in the spreading of the last-minute demand to several nights. Students planning to travel over the weekend took their exams on Friday or even Thursday. As a result, the day of heaviest demand became Thursday rather than the deadline day of Saturday.

Both of these administrative changes have resulted in almost no lines during the spring semester and in the following fall semester. However, lines did develop during the make-up periods. The first make-up period had small lines, the second had moderate lines and the third had huge lines. In spite of the fact that students were warned repeatedly not to wait for a make-up period, they did anyway. In fact, many students waited until the make-up period for their first attempt on all the examinations. To reduce this dependence on the make-up periods we are planning a further change: no bonus points will be given for examinations taken during the make-up period and only one attempt per exam will be permitted without penalty. This should discourage students from delaying beyond the initial deadline.

QUALITY CONTROL

The computer testing routine includes an item analysis procedure. After each semester this item analysis gives a discrimination index for each question. This is a point biserial correlation. It measures the relationship between the grade of each student and his or her tendency to choose the correct answer, a measure of the discriminating power of the item. The analysis also gives a difficulty measure with the percentage of students answering the question correctly.

Only after the 1983 fall semester were sufficient data accumulated to obtain meaningful reliability coefficients. In general, the reliability coefficients were very good for the set of questions on the examinations. The few cases of low reliability were due to two causes: (1) typographical error, and (2) fuzzy questions. Although the typographical errors are unfortunate, they can be expected in this huge set of questions (somewhere between 2,000 and 3,000). The important point to be made is that the item analysis procedure was able to detect typographical errors and, as a result, these typographical errors were corrected prior to spring semester.

Fuzzy questions are the result of poor question construction. Such questions were either re-written or eliminated. Thus, the examinations have been greatly improved, and will continue to be improved over time.

SECURITY

To maintain security, the student presents a picture ID to a test monitor, who assigns a workstation to the student. The student logs-on to the terminal takes the test, receives a grade, logs-off, and returns to the monitor to retrieve the ID. The monitor is able, at all times, to display on the monitor terminal a seating chart of all terminals and the test being taken by each student.

EVALUATION OF EXPERIMENTAL RESULTS

Although it was not possible to compare the results of computer-assisted testing via a controlled experiment, one can make certain generalizations based on observations and discussions with students and faculty. These will be grouped according to effects on students, on faculty and on administration.

a. Effects on students

1. More study and better learning.

   In general, students reported that they studied much more. With fixed goals (90% for A, 80% for B, etc.) they strived to achieve.

2. Less stress.

   Because there were no fixed times for the exams, because each exam covers only a small segment of the course and because the students may repeat an exam, the students find they can take the examination when ready under less stress.

3. Constant study.

With an exam almost each week, students tended to keep up with the material as presented in class. Under the old system of one or two midsemester exams plus a final exam, there were only two or three major cram-type study periods. In a course like statistics, students need to keep up or else the lectures become meaningless.

4. No cheating.

In a typical departmental exam with 400 or more students crowded in a room, it is difficult to detect and prevent cheating. In fact, a common complaint of honest students has been the existence of extensive cheating in departmentalized exams.

With randomized computer generated tests, it is essentially impossible for a student to cheat.

5. Better exam questions.

With the use of the item analysis feature, poor questions are eliminated from the test file. Also the type of questions that result in good reliability are noted and these can serve as a guide for future question construction. With better tests, we have more accurate measures of achievement.

6. Immediate reinforcement.

The computer grades each exam immediately and provides diagnostic advice to the student while the topic is fresh in the student's mind.

b. Effects on faculty.

1. Less test preparation.

Obviously, once a pool of questions is generated, there is no need for a rush job of mid-term and final test construction. This frees the faculty to spend more time helping students, preparing lectures and doing research.

2. No cheating concerns.

Cheating by students is always a concern of faculty.

3. Less personalized teaching.

The tests are obviously predetermined. This fixes the syllabus and test questions for each instructor. Thus, the instructor has little or no freedom of choice of material to cover. Also, enrichment materials are not particularly welcome to students if these topics are not covered on the exams. Many instructors do not like such a restriction.

On the other hand this may be the most desirable way to handle core courses, where the content affects future courses. There is a real need for consistency. Secondly, with less need for planning and scheduling, each instructor has more time to be creative in the classroom, to work with students and to do research.

There is nothing to prevent the instructor from basing part of the course grade on class participation, pop quizes, term papers, homework assignments, etc., in addition to the computer tests. In our course the entire course grade was based on the computer tests, and as a result students who thought they could learn through self study stopped attending classes. Class attendance near the end of the course dropped to about 40% despite the best attempts of the instructor to make the classes useful, interesting and informative. Most students who continued to attend classes reported that class attendance had a positive effect on their performance.

c. Effects on administration.

1. Fixed cost factors.

The hardware costs are lower when good management techniques are used to distribute the testing load uniformly over more hours. If student demand is uniform over the hours of open lab time each week, fewer terminals are needed than if all of the student demand occurs at a weekly deadline. With our testing policies we found twenty-two terminals were adequate to service 750 students who averaged 40 minutes per exam. Our testing lab was open 30 hours per week and was utilized at about 50% of capacity overall. With only ten terminals the lab would need to be open more hours per week, and some form of scheduling of students to lab times would need to be initiated. An alternative plan is to print out individual exams, have students work the exam away from the terminal, and then use the terminal only to enter the answers.

This results in less connect time per exam, and therefore fewer terminals required, but has an additional dependence on a mechanical printer which traditionally is more likely to break down. Also, the cost of paper becomes rather significant.

2. Variable cost factors.

   By going to a paperless exam procedure approximately one dollar per student was saved in paper and reproduction costs each semester. Furthermore, the exam monitoring was done by student assistants instead of the traditional course instructor, thus downgrading the cost of administering the exams. Hardware maintenance costs offset the savings somewhat.

3. Marginal cost factors.

   The marginal cost of adding another course to the system consists only of the cost of staffing the additional lab hours. Since Psychology had the hardware in place for their course, Business merely shared the computer maintenance cost and supplied the student assistants for the additional lab hours.

It appears overall that the computerized testing program has significantly more pros than cons. The final evaluation will come when these students take subsequent courses which use statistics as a prerequisite. Will there be a noticeable gain in statistical knowledge when these students take advanced courses in their junior and senior years? Initial reaction from instructors of courses that use statistics indicates that the general statistical level of the students has gone up.

THE FUTURE TESTING SYSTEM

The highly successful experience which the College of Business has had with computerized testing in the Psychology Department has made clear the value of this type of test delivery. But, the overload which Business has created on the Psychology facilities, the opportunity to expand the testing program to other business courses, and the need to make significant modifications and improvements to the testing system, suggest that the College of Business develop its own hardware and software for computer-based testing. This section describes the proposed design of a College of Business testing center.

Two major modifications are planned for the College of Business testing center, both of which should improve performance and permit significant expansion of the system. The first modification is the hardware configuration. A central minicomputer will be used for the storage of the question bank and for student record keeping. This minicomputer will be netted to a large number of microcomputers. The student logs onto one of the microcomputers. A test for this student is generated on the minicomputer and then the test is downloaded to the microcomputer, along with sufficient software for administering and grading the test. The student takes the test on the microcomputer and receives a grade. The student can even review the test results on the micro. After the student logs-off, the grade along with the student's results are filed on the mini. This downloading of tests to microcomputers will permit almost unlimited numbers of students to take tests simultaneously without interfering with each other on the central computer.

At the present time we are planning the testing center around a VAX 11/780 mini computer with an initial setup of 64 Rainbow microcomputers and a LA 100 printer. The Rainbows will be netted to the VAX. The VAX will download software and test materials to the Rainbows. The students will take the examinations on the Rainbows and upon completion of the test the results will be uploaded to the VAX for subsequent recording and analysis.

A second major modification is in software. Specifically, we plan to incorporate the latest psychometric tool of item response theory (IRT). Item response theory is probably the most significant development in psychometrics in many years. In item response theory examinations are tailored to the ability level of the student. This will most likely reduce the testing time and at the same time make possible item and test statistics which are dependent neither on the characteristics of the examinees nor on the other items in the test. These are invariant. With the item statistics it becomes possible to describe in precise terms the characteristics of the test before the test is administered. This capability allows one to construct a test that is highly efficient in accomplishing the purpose of the test.

Item response theory also provides an extremely powerful tool for special studies such as those conducted by marketing departments of business schools. Because of the resistance of individuals to being quizzed in marketing surveys, an efficient procedure is required. Thus, this testing center could provide a powerful tool for research in many areas of business.

Other modifications that are planned for the College of Business testing center include the review of the last test results by the student and his instructor, something that does not exist at the present time, a process of permitting students to take practice examinations at home or in the dorm, a "desk calculator" whereby a student can make his calculations during the test using the computer, additional use of graphics. It is expected that the testing center will be an extreemly useful center of testing.

CONCLUSION

The Texas Tech experiment with computerized testing has demonstrated that such a test delivery system is both practical and desirable. It is an ideal way for testing in monster-sized classes. It is a convenient way for incorporating the latest of psychometric testing techniques. It is the way most testing at a university probably should be conducted, and thus our experiences at Tesas Tech University, both past and future, will probably be of major significance.

REFERENCES

1. Conover, W. J. and Iman, R. L., Introduction to Modern Business Statistics, John Wiley and Sons, Inc., New York (1983).

2. Lord, F. M., and Novick, M. R., Statistical Theories of Mental Test Scores, Addison-Wesley (1968).

3. Warm, T. A., A Primer of Item Response Theory, National Technical Information Service, U. S. Government Printing Office (1978).

CREATING COST EFFECTIVE CAI FOR THE COLLEGE CLASSROOM: THE ROLE OF TEMPLATED AUTHORING AIDS

by

Boyd Richards
Hazeltine Corporation

CREATING COST EFFECTIVE CAI FOR THE COLLEGE CLASSROOM: THE ROLE OF TEMPLATED AUTHORING AIDS

Boyd F. Richards
Hazeltine Corporation

The computer revolution is finally infiltrating the college classroom. Every year, more faculty realize they can use computers to do more than teach programming or "computer literacy." They realize they can use computers to supplement regular classroom activities with extended drills, remediation, and self-paced instruction and to deliver whole courses such as introductory foreign language training. Professors of music, biology, chemistry, language training, and mathematics all across the country have helped to lead this revolution that is now attracting professors in the arts and social sciences.

However, we have a long way to go before the benefits of computers in the classroom are thoroughly attained. Three problems stand in the way: (1) purchasing the computers, (2) knowing how to use them best, and (3) reducing production costs. Fortunately, with the technological and theoretical advancements over the past years we have greatly reduced the first two problems. The cost of computers is coming down from around $50,000 per learner station in the 60's to around $3-4,000 now and should decrease further. Developments in higher education and industry are generating a rich groundwork of theory and practical advice about when and how computers should be used.

Although solutions to the problem of production cost have lagged behind the advances in hardware and theory, three different ways of handling the problem have recently emerged. The first solution calls for replacing programming languages with user-friendly production tools geared to non-programmers. These production tools, called authoring aids, range from high-level authoring languages to special-purpose editors that lead to a singular kind of lesson format. Another possibility, very different from the first, would have us avoid most production tools and instead work to improve programming expertise. Those pushing this solution feel that authoring aids impose too many constraints. And finally, we could move the burden of production from faculty to professional design centers. Only in this way, many argue, can the growing body of knowledge about CAI be properly applied. Most faculty do not have the required resources and time to properly undertake a project to design lessons on computers, just as they usually don't have the resources and time to write their own textbooks and films.

While I expect to see advances in all three cost cutting areas, at present I believe that authoring aids have the most potential for getting cost-effective CAI into the classroom. Too few faculty are ready or able to take upon themselves the burden of learning a programming language. On the other hand, we cannot count on commercial or professionally produced courseware (instructional software) because the K-12 market is currently much more attractive and development centers lack funding to write lessons for faculty. In other words, unless the faculty themselves take the initiative, the infiltration of CAI into their classrooms will proceed much too slowly.

With the ease of authoring that new aids appear to afford, faculty have the ability to create such things as individualized CAI lessons, on-line tests, sophisticated drill and practice activities, enrichment exercises for high achieving students, and simple simulations. I have seen these types of courseware developed and used with very positive results.

Because I have high hopes for what authoring aids will help faculty to achieve I have chosen to focus on their use in this paper. I begin by describing the major types of authoring aids and how they can be made easy and effective to use. I then illustrate ways in which faculty can use the aids to get courseware into their classrooms. I conclude with a list of suggestions faculty should consider before authoring computer lessons.

Types of Authoring Aids

Any computer-based software that increases the quality, productivity, or ease of creating instructional materials can be considered an authoring aid. While some computer-based authoring aids exist for creating instruction not delivered by computer, I will exclude them from my discussion. Furthermore, a growing number of aids exist primarily to facilitate design-related activities such as task analysis and the ordering of objectives. One such package is in planning by the Hazeltine Corporation for use on the MicroTICCIT system. Current plans call for design aids to facilitate organizing and retrieving data from task analyses, arranging learning hierarchies, and selecting media to use. But because my focus here is on reducing production costs, I will not deal with design tools in the paper. Nevertheless, I have very high hopes concerning their contribution to faculty who design their own CAI.

In this paper I will focus exclusively on computer authoring aids that directly influence the quality, productivity and ease of producing (i.e. entering into the computer) instruction on computers. And even seen these types of courseware developed and used with very positive results.

Because I have high hopes for what authoring aids will help faculty to achieve I have chosen to focus on their use in this paper. I begin by describing the major types of authoring aids and how they can be made easy and effective to use. I then illustrate ways in which faculty can use the aids to get courseware into their classrooms. I conclude with a list of suggestions faculty should consider before authoring computer lessons.

Types of Authoring Aids

Any computer-based software that increases the quality, productivity, or ease of creating instructional materials can be considered an authoring aid. While some computer-based authoring aids exist for creating instruction not delivered by computer, I will exclude them from my discussion. Furthermore, a growing number of aids exist primarily to facilitate design-related activities such as task analysis and the ordering of objectives. One such package is in planning by the Hazeltine Corporation for use on the MicroTICCIT system. Current plans call for design aids to facilitate organizing and retrieving data from task analyses, arranging learning hierarchies, and selecting media to use. But because my focus here is on reducing production costs, I will not deal with design tools in the paper. Nevertheless, I have very high hopes concerning their contribution to faculty who design their own CAI.

In this paper I will focus exclusively on computer authoring aids that directly influence the quality, productivity and ease of producing (i.e. entering into the computer) instruction on computers. And even more specifically, I will focus on aids that either by careful design or because of built-in limitations template the production process.

Two main types of authoring aids are available: authoring languages and authoring systems.

Authoring Languages. Authoring languages are specialized, high-level computer languages containing commands for simplifying constructing displays, analyzing student responses, and providing simplified branching. Some examples of authoring languages are TUTOR used on PLATO, CAMIL used on McDonnell Douglas's AIS, and SUPERPILOT used on many popular microcomputers. As with programming languages such as BASIC, FORTRAN, and PASCAL, authors program with authoring languages by writing commands in order of execution along with loops and conditionals. However, unlike programming languages, authoring languages have commands to specify, for example, which student answers could be judged equivalent or how to juxtapose text and graphics.

I know very few faculty who by themselves have created CAI with authoring languages. Although usually simpler than programming languages, their use still requires many programming skills. Consequently, most faculty will still require the help of an experienced programmer.

Just as programming languages use different formats and have different capabilities, so do authoring languages. On one end of a continuum, as shown in Figure 1, we have very sophisticated languages like ADAPT Level 3. ADAPT Level 3 (Hazeltine's authoring language for MicroTICCIT) can support development of everything from simple drills to complex, multi-terminal simulations. On the other end of the continuum, we find very simple languages such as PILOT used on the APPLE. Among other things, as we move from the left to the right along the continuum, the languages become more restricted. That is, there are more constraints (i.e., things the languages cannot do), and there are more rigid formats that must be followed (P. Merrill, 1981). But they are easier to use.

<--->
ADAPT 3 PILOT

Figure 2.

Authoring Systems. Authoring systems consist of software programs typically dedicated to a single type of computer. Among other things, these programs provide authors with on-line editors (i.e., special wordprocessing packages) to use in specifying the content and lesson structure.

The MicroTICCIT system helped to pave the way for authoring systems and continues to be a major trend setter. When TICCIT was developed almost 15 ago, it contained one of the first, yet one of the most versatile authoring systems. This system, known as APT (Authoring Procedure for TICCIT), not only cut production costs by eliminating programming, it improved designers' efficiency by providing a theoretically-based design format (Merrill, M. D., Schneider, & Fletcher, 1980). Using APT, faculty can create instruction that is segmented into 3 types of presentation forms: rule statements, examples, and practice questions. Built-in branching specifications allow learners to select these components in any order and keep the author from having to program the branching. To accommodate both fast and slow learners, APT allows faculty to write elaborated and concise variations of rules, include helps during practice, give advice about learning strategies, and systematically vary the difficulty of examples and practice questions.

Several years of intense development by groups around the country demonstrated the effectiveness of APT as an authoring aid. Courses in algebra, English, telecommunications, aircraft flight procedures and many foreign languages have been created by universities, the government, and industry with very efficient use of resources. Because of the success of APT and the inherent appeal of authoring aids that don't require programming, many small authoring systems are now available for microcomputers. These systems are smaller than APT because they tend not to contain features such as built-in course structures, learning strategy advice, data collection procedures, and student performance reports. Table 1 lists some of the new small authoring systems that are available.

Table 1. Sample small authoring systems

Name	Used On	Sold By
AIDS	Apple II	Skillcorp Software, Inc.
The AUTHOR	IBM PC	Phoenix Performance Systems
CAIWARE	TRS-80	Fireside Computing, Inc.
GHOSTWRITER	Apple II	CAURI Interactive Video
IAS	IBM PC	McGraw Hill
PASS	Apple II	Bell & Howell

Authoring systems tend to rely on one of three techniques to simplify the production process: prompts, menus or syntax. Prompts are questions asked by the on-line system which guide authors through the process of specifying text, graphics, branching and answer judging. Menus are lists of options authors can select one at a time. Each option allows authors to give important information about the emerging lesson. Syntax formats use on-line blanks which authors fill in according to a specified format.

Templating the Authoring Aids

In general, authoring languages and systems help reduce production costs by simplifying the demands of entering lessons onto the computer. These reductions in cost can be increased by adding templates to either authoring aid. Templates are content-free lesson structures or shells. They provide the slots and scaffolding into which authors can insert the content with the authoring language or system (and possibily a programming language). The purpose of these templates is to give authors as many ready-made decisions as possible--to simplify as much as possible their task of specifying text, question formats and branching.

There are three types of templates commonly found in authoring languages and systems. The first type, characterized by TICCIT's APT, provides a general format that prescribes the structure and instructional design approach of lessons. When these templates are consistent with instructional theory they not only help reduce production costs, they also help authors design their lessons effectively. They guide authors by dictating how to break the content into presentation forms and how to present them to students. The second type of template is much narrower in scope and specifies the format for screen layouts (i.e, position and amount of text and graphics) and often dictates the sequence and number of different types of screens (e.g., introductions, rule statements, questions, etc.).

While the first type of template allows many things to vary from lesson to lesson such as the number of pages of rule statements, size and position of graphics, scoring, sequencing, and number of practice questions, and overall page layout, the second type contains very rigid slots for frames of specific types of text, graphics or student input. The second type of template further specifies much of the format of each frame. The difference between the two templates is analogous to blueprints for prefabricated houses with different types of customer options. APT is like a set of blueprints that buyers can mix and modify to customize their prefabricated house so that every house may be similar in style but unique in many respects (e.g., size, number of rooms, type of fireplace). Smaller templates are like blueprints with a few superficial options such as color of carpet and type of wallpaper.

The third type of template is a "homemade" master lesson that serves a as model for other lessons. They are created with a programming language, authoring language or authoring system just as any lesson would be. In fact, the template may even be an actual lesson that authors copy to create a new lesson. They replace the content in the copy of the master lesson with the desired content for the new lesson. Authors can also alter, delete, or add commands or specifications in the new lesson so that the final lesson may end up looking very different than the original template.

With the first two types of templates the lesson format and structure are mostly provided by the computer when authors insert content into them. With the third type of template, the format and structure are copied from the master lesson and then can be modified.

Using Templated Aids Effectively

Whether a template will help faculty create cost effective CAI depends on several things. One major condition is the ability of the authors to use the language or system for which the template was created. Another is the match between the constraints of the template and the needs of the author. Frequently, authors need more flexibility than the template permits. This is particularly true of templates that dictate the position and amount of text on each screen. A third factor is the quality of the template itself. Templates based on arbitrary decisions about what lessons should look like are not as likely to help faculty as are templates based on instructional theory. With theory-based templates that meet an author's needs, the authoring process is reduced to three steps: identifying the content, breaking the content into pieces as prescribed by the template and inserting the pieces into their proper places. Unfortunately, most templates are not based on acceptable instructional theories and impose too many arbitrary constraints. At times, even good templates do not fit the authors' situation and would interfere in the design or production process.

The problem of finding an effective template is further complicated by the demands of the host language or system. For example, templates in authoring languages demand too many programming skills for nonprogrammers. Templates in authoring systems may be too simplistic or restricted for programmers.

Fortunately, a new breed of hybrid authoring aids are becoming available that increase the likelihood of being able to use the best template at the right time. One of the best examples of a hybrid is ADAPT, the new authoring package for MicroTICCIT. When TICCIT was first created, authors only had APT, the templated authoring system. Recall that APT required all TICCIT instruction to be neatly organized by objectives into rule statements, examples, and practice and that the learner be allowed to select these pieces in any order. While a lot of excellent, cost-effective instruction was created with this templated system, authors soon found that for certain kinds of instruction it was too restrictive. TICCIT developers at Hazeltine responded by providing a very flexible, non-templated authoring language called TAL. Now authors had the freedom to create many varieties of instruction. But a new problem arose. TAL required programming skills, as do all other high-level languages.

ADAPT, still under development, combines the initial templated system with the versatile authoring language into a multilevel environment (Mudrick, 1984). Level 1 consists of simple prompts and menus for first-time users and for the development of straightforward courseware. As their CAI authoring ability increases, authors can move to higher levels. Level 2 contains more complete menus of available options. Level 3 is the authoring language. Apart from the levels in authoring format or tools, authors can decide what underlying template, if any, to use. This means that an author moderately skilled in programming could choose to create a lesson using the menu-driven system with the rule-example-practice template. An advanced programmer might choose the authoring language with no template. In fact, when the moderately skilled author is ready to move from menus to commands, the lessons created at the one level can be moved up to the higher level without any loss or duplication of work. In a similar manner, authors can use different tools as they improve their ability to write quality lessons. Initially most authors would want to rely heavily on the templates. In time, as their skills improved, they could gradually move away from the templates to write customized lessons. In fact, a lesson created with a template can be easily modified so that any or all parts no longer adhere to the template's formats.

Why Templated Aids

Creating good CAI is difficult; it takes time and careful planning. Nevertheless, I believe there are faculty who can and ought to become involved. I disagree with some of my colleagues who suggest that creating a lesson on computers is like creating a textbook and that faculty don't have the time, skills, or resources to undertake such a project. On the contrary, I suggest that with proper use of templated authoring tools, the size of the task can be reduced to manageable levels. Nearly all aids will reduce the burden of inputting the lesson on the computer. Aids such as ADAPT will also reduce the burden of writing the initial lesson.

I have seen numerous faculty become successful authors either by themselves or teamed with an instructional designer or programmer. One professor I know spent 3-4 hours a week for several months creating German drills on the APPLE computer. He knew little about computers or instructional design. But with a little help from a student consultant he created a simple template using PILOT and used it to create several cost effective lessons. His students can now use the lesson to drill vocabulary words on their own.

Faculty at Brigham Young University, Northern Virginia Community College and Phoenix Community College have efficiently created lessons on foreign language training, basic algebra, writing, and numerous other areas on TICCIT using APT and ADAPT.

PLATO contains many hours of instruction developed by faculty in nearly every subject area. Although they created most of this instruction with TUTOR, PLATO's authoring language, some was created using templates. I was involved with several faculty from Indiana, for example, who developed a template for simple simulations or dialogues between library workers and patrons. Using this template we created numerous lessons at very little cost to teach use of the card catalog.

Other faculty at several universities have used a template called the Corrective Feedback Paradigm (CFP) to teach paired associates (Siegel and Misselt, 1984). I have seen it used to teach the bones of the body and foreign language vocabulary words. Even though use of the CFP results in very sophisticated drills with complex scoring, sequencing, and repetition of missed questions, all the author has to do to create a drill is type in the list of paired associates.

Research evidence illustrates the effectiveness of authoring aids. By observing 143 production groups, over an extended period, Avner, Smith and Tenczar (1984) found that use of authoring aids helped to predict productivity:

> Software production tools showed good results both in original and export settings though documentation inadequacies frequently decreased effectiveness in export settings. Software production tools ranged in sophistication from templates (sometimes simply a CBI lesson that was revised for each new application) to complete CBI languages and authoring environments. (p.88)

But authoring aids are not without problems. I have already mentioned the usual loss of flexibility. In addition, authoring aids tend to preclude creating instruction that can be moved to different cannot ignore the importance of hardware. Clearly, deciding to use ADAPT does no good unless the author has access to a MicroTICCIT system because ADAPT only runs on TICCIT and MicroTICCIT. So until a MicroTICCIT system is purchased, aids for available hardware would have to be used. Computer software catalogs, magazines, and user groups will provide most of the information needed about the tools for computers commonly used in higher education.

When hardware isn't already available but will be purchased, the availability of a range of authoring aids, templated and non-templated, should be examined and used to decide which computer to buy.

2. Aids are a means to an end, not the end itself. A child given a hammer will look about for things to pound; a professor with a good templated aid may be tempted to search about for content to be inserted in it. Using the template, like the hammer, becomes the goal. However, such authors must not forget that the primary goal of creating CAI is the improve classroom instruction.

3. Templated aids reduce but do not eliminate production costs. No matter how efficient or easy-to-use a templated production tool may be, it is impossible to avoid all production costs. In fact, many times using a templated aid merely trades one cost for another. For example, courseware templates can save a lot of programming time but may require a lot more time searching for, or creating an acceptable template.

4. Select a template carefully. Authors need to carefully match their CAI and design abilities and ambitions with the available templated authoring languages and systems so that they are not drowning on the one hand or trying to swim in a mud puddle on the other. The value of both types of templates depend on the the author's needs and

types of computers. This means instruction produced on PLATO cannot currently be run on an APPLE and vice versa.

Another disadvantage of authoring aids, except for full-range software like ADAPT, is that they do not allow room for growth. Many authoring languages and their templates are too complex for non-programmers. However, many authoring system and their templates are too simple for programmers. This means non-programmers who become programmers most often learn how to use a templated authoring aid only to put it aside after they have acquired programming skills. At first, prompts, menus, restricted authoring languages, and templates cut costs by helping novices get started and by taking care of a lot of the work to make a given type of lesson. In time, however, as the novice evolves into an expert and the singular lesson format is no longer acceptable, many of the constraints in these tools tend to interfere and slow down the process. ADAPT allows you to "get rid" of prompts and bypass menus.

The Role of Templated Aids

Not all faculty should create CAI. Likewise, not all college-level CAI should be created by faculty. However, many faculty will need to create much of the CAI if computers are going to be properly used in higher education to support on-going instruction. And finally, the professors that do undertake a project to develop instruction on computers ought to carefully consider how templated authoring aids can help them. The following list of suggestions about the role of authoring aids reflect the kinds of things to think about.

1. Aids are secondary to hardware. Because most authoring tools can only be used on a specific computer, faculty considering aids

project requirements. However, whenever possible, theoretically sound templates should be used. These templates have more probability of leading to successful instruction.

5. *Regardless of the aids being used, good lessons require good design*. Most authoring aids, especially templated ones, contribute to the design process (i.e., planning) by providing ready-made lesson formats and specifications. However, this does not allow authors to ignore the design phase completely. Correct content must be found and broken into chunks and the importance and sequence of the chunks must be established.

6. *Use of aids should be supplemented with expert advice*. Few faculty have received explicit training in designing instruction, especially CAI. Through experience they have learned many personal approaches and techniques. These will not often work on computers. To avoid creating instruction that doesn't work, faculty should consult wherever possible with colleagues, computer specialists and instructional designers. In fact, students could even be hired to do the bulk of the production. Just because production tools are available, faculty do not have to do everything themselves. Many projects I have conducted using production aids were greatly enhanced by the involvement of others.

7. *With help many aids can and should be customized*. Avner, Smith and Tenczar found, for example, that the best lessons were often done with tools that had flexibility (i.e., weak templates), and could be modified to meet a specific situation. I also found this to be true while using the TICCIT authoring language to develop basic skills lessons on the TICCIT computer. I used it to form a template that then

served as the backbone of every lesson.

6. *Look to the future and expect some exciting things to occur.* We are at the frontier of authoring aids. Although much has been done, a lot more is yet to be discovered or developed. New and better templated tools are always on the horizon.

References

Avner, A., Stanley, S., and Tenczar, P. (1984). CBI authoring tools: effects on productivity and quality. *Journal of Computer-Based Instruction*, *11*, (3), 85-89.

Kearsley, G. (1984). Authoring tools, an introduction. *Journal of Computer-Based Instruction*, *11*, (3), 67.

Merrill, M. D., Scheider, & Fletcher, K. (1980). *TICCIT*. Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, P. F. (1981). The case against PILOT: The pros and cons of computer-assisted instruction, computer languages and authoring systems. Paper presented at the annual meeting of the American Educational Research Association, Los Angeles, CA.

Mudrick, D. (1984). Adaptive authoring systems for computer-based instruction. *Journal of Computer-Based Instruction*, *11*, (3), 82-84.

Siegel, M. A., & Misselt, A. L. (1984). An adaptive feedback and review paradigm for computer-based drills. *Journal of Educational Psychology*, *76*, 310-317.

Young, J. L. (1984, October). The case for using authoring systems to develop courseware. *Educational Technology*.

UTILIZING THE MICROCOMPUTER AS A TAX PLANNING TOOL IN BUSINESS AND EDUCATION

by

Stan Scott

Peat, Marwick, Mitchell & Co.

Utilizing the Microcomputer as a Tax Planning Tool in Business and Education

by Stan Scott, Senior Manager

Peat, Marwick, Mitchell & Co.

The Peat Marwick tax departments have used computer resources in their work for almost fifteen years, and the microcomputer for the last several years, when the IBM Personal Computer (IBM PC) first became available. This speech discusses the various uses to which these resources have been put, both in business and education, and their impact upon our tax practice.

I. History of computerization in the Peat Marwick tax departments

In early 1970, Peat Marwick began using the General Electric timesharing network. Through the use of timesharing terminals, Peat Marwick personnel had access to hundreds of programs, databases and various information networks, such as U.S. census data.

This network had two principal uses: First, it provided a nationwide electronic mail network, which has been used for such purposes as inter-office communication and up-to-the-minute bulletins on new tax legislation, such as TEFRA in 1982 and the Deficit Reduction Act of 1984. Second, Peat Marwick developed many tax programs for on-line use on the General Electric system, including PLAN1040, a complete individual tax planning system, PLAN1120, a corporate tax planning model, and BANKHOLD, for use with one-bank holding companies.

In late 1981, the Peat Marwick tax departments adopted the IBM PC as their standard microcomputer tool, and by the summer of 1982, each of the approximately 100 operating offices in the United States had at least one IBM PC. These computers are used both on a stand-alone basis, running a wide variety of software, and as communications devices to connect with mainframe computers for use with such applications as tax research and electronic mail.

The advantages of using a microcomputer-based system over our previous timesharing system have been tremendous. First, the cost savings through the elimination of timesharing charges, including both storage costs and access charges incurred by the Firm in using the available programs, have been enormous. Second, because our tax applications are, by and large, spreadsheet based (a point I will discuss more fully later), it has been far faster and easier to write new tax applications than it had been using programming languages such as FORTRAN or COBOL. Spreadsheet-based tax software may also be updated for tax law changes in much less time, and these applications may be easily modified by the operating offices to fit their particular needs. And, finally, there has been an explosion of inexpensive tax software written for the microcomputer, in such areas as estate planning and depreciation.

We have also found that the IBM family of microcomputers has been ideal for our tax department's needs. First, the IBMs have the computing power we need, which includes both faster calculation and disk access time. Second, IBM's presence in the microcomputer world means that there has been a whole host of peripherals available, including add-on memory boards, printers, and color boards and monitors. Third, and most importantly, there has been an enormous amount of software written for the IBM, including both generalized productivity tools, such as spreadsheets and word processors, and specialized tax applications.

The administration of microcomputer activities in each operating office has been handled as follows. Each office has a professional designated as a Tax Computer Applications Coordinator (TCAC), who handles such administrative matters as disseminating information received on new programs and techniques from Executive Office in New York, and a Tax Computer Specialist (TCS), who is perceived as a focal point for microcomputer expertise in the office. Each TCS receives a week of intensive training on microcomputer topics, which is discussed later. For the last two days of this training, the TCSs are joined by the TCACs for discussions of administrative matters as well as current developments in the microcomputer area.

II. Use of the microcomputer in the Peat Marwick tax departments

The use of the microcomputer in Peat Marwick tax departments has focused on four areas: tax calculations, planning models, research tools, and education. Because the calculation of tax liability for individuals, corporations, and other taxable entities is frequently an involved, time-consuming process, we use a wide variety of off-the-shelf tax packages as well as software we have developed ourselves.

As an example of outside software in use by Peat Marwick, there are several comprehensive depreciation packages available on the market today, allowing our clients to avoid the high costs of timesharing or manual upkeep of depreciation schedules, especially where different book, state, and federal methods are selected. Another area is that of comprehensive financial planning, which brings together planning techniques in individual income taxation, estate and trust planning, insurance, etc. to set up an overall financial plan for an individual.

Most of Peat Marwick's tax packages have been written for Lotus' 1-2-3 system, and have been rewritten to operate with the Lotus Symphony software. This program allows us to write sophisticated tax packages quickly and efficiently, which may be easily updated for tax law changes as they become applicable. Symphony has allowed us to provide an on-line help feature for our individual and corporate planning models, so that a user manual is available as an integral part of the system. These packages can be modified by users to fit their particular needs, as, for example, where state tax calculations are included in our federal individual tax model. And, finally, these applications provide ideas and programming techniques for our professionals to use in their own applications.

Peat Marwick 1040, our individual tax planning model, was developed almost three years ago. At that time, it was written for the VisiCalc spreadsheet, which at that time was the most powerful spreadsheet available for the IBM PC. Over the past three years, it has been rewritten for Lotus 1-2-3, and, most recently, Lotus' Symphony program, to take advantage of such features as windowing and an on-line help feature.

Peat Marwick 1040 performs all major tax calculations for taxable years 1982 and forward. It can calculate up to five different alternatives simultaneously, permitting the user to investigate the impact of one or more planning alternatives

186

for a particular taxable year, such as income averaging and tax shelter analysis, or to track the tax liability of an individual for up to five consecutive taxable years. In addition, the model can be customized for the user's particular needs, such as state tax calculations, cash flow projections, and other relevant reports.

The following are just a sample of the calculations performed. Peat Marwick 1040 calculates the two-earner married couple deduction, the capital gain exclusion, limitations on such items as investment interest, medical expenses, and charitable contributions, and special taxes, such as the alternative minimum tax, the special averaging tax on lump sum distributions, and self-employment tax.

Output is provided in the form of summary reports, supplemental schedules providing detail for such items as the calculation of the allowable investment interest deduction, and a graphic representation of tax shelter income or loss, taxable income and tax due. These graphic summaries have proven to be highly effective in client presentations, because they provide a quick pictorial summary of a client's income tax situation.

This model may be used in a variety of different ways. It may be used for a taxpayer at several times during the year to calculate his estimated tax liability, to ensure a safe estimate. It is used for year-end planning purposes, to focus on such decisions as whether to take a capital loss in the current year, or to wait until a subsequent year. And, finally, it may be used to check the tax computed independently by an outside tax return processor such as Fast-Tax or Computax.

Peat Marwick 1120, which also operates with Lotus' Symphony program, is similarly designed for corporate tax planning. Like 1040, Peat Marwick 1120 performs all major corporate tax calculations for taxable years 1982 and forward, for up to five alternatives at a time.

For regular corporations, 1120 calculates such items as the following:

- Capital gain net income
- State taxable income
- State income taxes
- Federal and state charitable contribution limitations

- Net operating loss deduction and carryover
- Dividends received deduction
- Federal taxable income and regular income tax
- Alternative tax
- Fiscal year tax computations
- Variable surtax exemptions
- Various credit calculations, limitations, and carryover, including the foreign tax credit, investment tax credit, and research and experimentation credit
- Capital gains preference amount
- Add-on minimum tax

Because Peat Marwick has a substantial client base of financial institutions, including both banks and savings and loans, Peat Marwick 1120 includes comprehensive calculations of bad debt deductions and bad debt preference amounts. For banks, 1120 calculates the allowable bad debt deduction under both the percentage and experience methods. For savings and loans, 1120 performs four calculations, the percentage and experience methods also applicable to banks, as well as the deduction allowed under the percentage of taxable income method and the experience method for nonqualifying loans. For all entities, 1120 will calculate the minimum or maximum bad debt deduction allowed, and will calculate the bad debt preference amount.

The use of this corporate tax calculation package has all of the advantages outlined with Peat Marwick 1040. The tax professional is encouraged to focus on planning techniques, with less emphasis on the tedium of working through the calculations.

There are a number of other tax calculation packages available to the tax professional. For example, penalty calculations for underpayment of estimated tax by individuals and corporations are performed by two tax planning models, Peat Marwick 2210 and Peat Marwick 2220. These models calculate the underpayment, test the applicability of any or all exceptions to the penalty, and calculate the penalty by applying the appropriate interest rates. Interest calculations on tax due or owing are provided by our TAXINT model, which provides detailed calculations going back through 1975, if desired, applying the applicable interest rate for each period of underpayment, and summarizing all interest due.

At times, it is necessary to perform tax related calculations to determine whether a particular set of tax rules apply, for example, in the area of personal holding companies. Peat Marwick PHCO assists the user in determining whether personal holding company status exists, by calculating ordinary gross income and adjusted gross income and applying the special rules for the determination of rental and royalty income as personal holding company income. If it is then determined that personal holding company status exists, PHCO calculates the undistributed personal holding company income and the tax due.

In addition to tax calculation models, a number of Peat Marwick microcomputer packages assist the user in corporate tax planning, in such areas as inventory analysis, asset purchase vs. lease analysis, and cash flow projections. I will discuss some of these packages briefly.

Peat Marwick LIFO is an inventory model that calculates end-of-year LIFO inventory values for up to thirty years, and computes the LIFO reserve by comparing LIFO to FIFO values. Comprehensive reports are produced, including a summary report showing FIFO and LIFO inventory values and the cumulative LIFO reserve, the composition of LIFO layers using base year costs, and the composition of LIFO layers using indexed costs. As with many of our other planning models, the program produces a graphic representation of the summary report, which may be highly effective in client discussions.

Peat Marwick CASHFLOW is a planning model that produces loan amortization and cashflow reports for tax years 1981 and forward. For loans, this model provides a detailed loan amortization schedule under a variety of assumptions, such as interest only loans, constant principal loans, balloon payments, etc. Comprehensive investment tax credit and depreciation calculations are also produced, including the selection of full or reduced ITC rates under TEFRA, use of ACRS or straight-line depreciation, and the immediate expensing of a portion of the asset cost under I.R.C. Section 179. Similarly, Peat Marwick BUYLEASE performs comprehensive depreciation and loan amortization calculations, to determine whether the purchase or the lease of an asset produces the most favorable results.

When a bank wishes to set up a one-bank holding company, various financial guidelines set out by the Federal Reserve Board, concerning such items as required capital to asset ratios, must be met. Peat Marwick BANKHOLD facilitates this analysis by comparing the impact of a whole range of financial assumptions, including the effects of a beginning asset or capital change, a change in the percentage of bad debt deduction or asset growth, or alternative ownership percentages. The model produces comprehensive reports of projected assets, earnings and capital accounts, the projected cash flow as a result of the proposed acquisition of bank stock, the projected holding company balance sheet as a result of the proposed stock acquisition, and a schedule of unpaid balance on long term. When the optimal results are obtained, these reports are included in the application forms filed with the Federal Reserve Board to obtain permission to set up a one-bank holding company.

This is only a sample of Peat Marwick's microcomputer capabilities in the area of tax compliance and planning. In addition, the microcomputer is used as a tax research tool, primarily with Mead Data Central's LEXIS system. LEXIS provides the tax professional with an on-line research capability, giving him access to current tax cases, regulations, revenue rulings and a whole range of other material.

III. Use of the microcomputer in Peat Marwick educational programs

In addition to the use of the microcomputer in tax compliance, research, and planning, the microcomputer is invaluable in the Peat Marwick professional training courses. In the first 18 months of service with Peat Marwick, every tax professional is expected to complete a self-study course in the use of the microcomputer. This course includes the following topics:

General housekeeping. The professional is trained in the general use of the microcomputer, including basic DOS (Disk Operating System) commands, such as how to copy a file, how to make floppy disk backups, how to use a printer, etc.

Use of Lotus' 1-2-3 and Symphony. Lotus provides us with an all-purpose tool, allowing our professionals to prepare sophisticated spreadsheets with no prior programming experience, communicate via our electronic mail system, and set up client databases. For this reason, the introductory course teaches basic Lotus commands, stressing the spreadsheet, since this is the most frequently used Lotus function.

Peat Marwick tax software. After the professional has mastered the rudiments of the Lotus system, he is trained in the use of Peat Marwick software. Although only a few of the many available packages are covered in this course, the knowledge gained carries over to our other packages, because of their uniform design. So far as possible, the menus, input screens, and report formats are uniform. This lets the professional concentrate on the tax calculations specific to the particular application, and minimizes the time necessary to learn how to use the package.

This introductory course provides the professional with a good working base on which to build in his subsequent use of the computer. Further, in our tax courses, the use of the microcomputer has been integrated wherever possible. For example, in our introductory tax course, which focuses on individual taxation, Peat Marwick 1040 is used with problem material to provide quick numerical calculations. This not only reinforces the knowledge gained in the introductory microcomputer course, but stresses the use of the microcomputer in the context of the client's overall financial and tax status and our importance in helping the client plan properly.

As noted earlier, each Peat Marwick tax department has one or more Tax Computer Specialists, who act as focal points for microcomputer expertise in their respective offices. The TCSs are given a week of intensive training in this area, which covers topics such as the following:

- Symphony training. Two and one-half days are devoted to intensive training on Lotus' Symphony. This course builds on previous knowledge gained of Lotus' 1-2-3, allowing the course to focus on advanced spreadsheet and database techniques, as well as modules on word processing, communications, and graphics.
- Peat Marwick tax software. Recent developments in Peat Marwick tax software are covered, such as enhancements to Peat Marwick 1040 and 1120.
- Spreadsheet techniques. Pointers are given on efficient spreadsheet techniques, such as grouping input items, setting up uniform menus, etc.
- Outside vendors. Current developments in tax return processing, estate planning software, and real estate planning are provided by outside vendors.
- Other topics. Other miscellaneous topics include the use of LEXIS for tax research, data transfers between the Apple Macintosh and the IBM PC, and microcomputer training techniques.

We expect that the knowledge gained in seminars of this sort will be put to use in actual client applications, which will in turn result in even greater use of the microcomputer in the operating offices.

IV. The impact of the microcomputer on Peat Marwick tax professionals

Although the microcomputer has been in use by the Peat Marwick tax departments for only a few years, the impact on our professionals has been profound. We have seen the following trends in this area:

The use of comprehensive tax calculation models have had a number of beneficial influences on the tax professional's attitudes toward performing tax calculations. First, the tedious "number crunching" involved in individual tax planning has been minimized, allowing the professional to focus on planning techniques. Second, the time required to learn the rudiments of taxation has been greatly reduced at the staff level. For example, because Peat Marwick 1040 combines all individual tax calculations on one worksheet, the staff person is immediately aware of the impact of the alternative minimum tax, the investment tax credit limitation, etc. Third, because our models encourage the user to calculate tax liability for a number of consecutive years, the professional is less apt to focus on minimizing the tax liability for a single year, while ignoring the overall impact on the taxpayer.

It is no longer the case that computer expertise is centered in only a few individuals with access to specialized computer programming abilities. The wide variety of microcomputer tools available, most importantly the spreadsheet, bring the power of the microcomputer directly to the professional. This means that the professional can attack a client problem, such as cash flow calculations, directly, instead of being at the mercy of a data processing department.

The microcomputer has also helped the professional to develop greater efficiency in other related areas. For example, with the word processing software now available, a professional can produce a draft report and make corrections in a fraction of the time previously required. The use of microcomputer databases have allowed us to offer clients new services in maintaining their records more efficiently. For example, a client's financial records can be kept on an electronic database so that monthly reports and year-end totals are quickly calculated.

In addition, because the microcomputer explosion is not limited to our practice, but to the business world at large, our clients are growing more and more sophisticated in the use of the computer. This means that we can, in addition to providing the results of our tax calculations, work with the client in developing his own models and planning packages.

More and more, our professionals have recognized that the microcomputer is not only an invaluable tool, but is central to our professional practice. With the pressures of time and the costs of professional time, we can no longer afford to waste professional time in unnecessary manual calculations. For this reason, our professionals are encouraged to use the microcomputer whenever feasible.

V. The future of Peat Marwick's computer capabilities

Not only is the microcomputer central to our tax practice today, but tomorrow's capabilities will extend and broaden our use of the microcomputer. Although it is difficult to predict future developments, we see the following capabilities on the horizon.

The first is in the area of integrated office automation, the so-called local area network (LAN). Such a network allows the microcomputers to run independently of each other, but have the ability to get information from other computers on the network, and use shared printers and disk drives. In addition, data entered on microcomputers can be transferred to dedicated word processing equipment, so that draft reports prepared by professionals can be transmitted to their secretaries for final editing and printing. We are now conducting pilot studies in several operating offices to test the feasibility of such systems.

The second is in the area of expert systems, which are sometimes referred to (incorrectly) as artificial intelligence systems. Such an expert system will contain a massive amount of tax information, including the Internal Revenue Code, regulations, rulings, and court cases, along with rules that make up the system's "inference engine." This information will allow the system to analyze data and make recommendations on the proper tax treatment of a particular situation, or suggest tax planning techniques.

This would take our microcomputer capabilities beyond the areas of tax and financial numerical calculations and tax research, and into decision support systems. At the present time, only pilot studies are feasible, because most current expert systems are simply too large and complex to function on microcomputers. However, as the technology becomes available, with a drop in costs, we will investigate this area further.

Third, we expect that enhanced training techniques will be available through use of the microcomputer. For example, the computer, through the use of questions and answers, might provide specialized tax training for the professional, geared to his particular interests and needs. This will allow our professionals to keep current as to new developments in taxation, as well as broadening their knowledge in the field.

SOME CURRENT AND PROJECTED USES OF COMPUTER TECHNOLOGY IN THE TEACHING OF STATISTICS

by

N. Scott Urquhart

New Mexico State University

SOME CURRENT AND PROJECTED USES OF COMPUTER TECHNOLOGY IN THE TEACHING OF STATISTICS

by

N. Scott Urquhart

Department of Experimental Statistics
New Mexico State University

ABSTRACT

Students in applied statistics courses present a number of instructional challenges: They come from diverse subject matter areas; many are afraid of symbols and formulas; some even have problems dealing with concepts rather than the memorization they often face in other courses. In a day when most routine statistical computations are available with hand-held calculators and more complex ones are available on desktop computers, students need to learn how to select an appropriate technique, and interpret the resulting analysis. To do this, should understand concepts needed to select an appropriate procedure, and they should gain experience in making decisions in contextually comfortable settings.

This paper describes three current and one developing use of computers which respond to some feature of the instructional setting described above. For the purposes of this conference, these uses should be regarded as ways to respond to instructional challenges, not as objects of interest.

1. INTRODUCTION

Did you take an introductory applied statistics course as an elective? Probably not. Applied statistics is a graduation requirement for most of our students, graduate as well as undergraduate. Required courses outside of students' majors have two strikes against them before they start.

This introduction expands on the instructional environment in applied statistics to give the setting for the four illustrations of computer enriched instruction. The illustrations range from fairly mature applications to ones in the initial planning stages; they use computers ranging from mainframe to microcomputers. Because this paper is intended mainly for nonstatisticians, it concentrates on the motivation for the computer usage and on its general form, not particularly on its technical details. Hopefully, this approach will help illustrate kinds of opportunities for using computer technology in instructional situations. The first two illustrations were discussed for statisticians in Southward, Urquhart, and Ortiz (1983).

Students in an introductory statistics course exhibit several forms of heterogeneity which should influence how we teach statistics. They come from a wide variety of disciplines: from accounting to zoology; from agriculture to engineering technology; from computer science to police science. They differ greatly in previous experience: The student who has gathered and tried to understand data has an entirely different perspective from one who has few real world experiences with data. Some, but not all, of this diversity can be removed by creating different sections for students with different backgrounds, but the utility of extensive sectioning is limited by scheduling and funding.

The national media have noted the general decline of ACT scores and similar measures of college preparedness. These declines include mathematics preparedness. These declines have led some institutions, New Mexico State University included, to impose some sort of mathematics proficiency requirement as a prerequisite for entry into courses past a specific level. Remedial courses exist to help the unprepared students meet the requirement. In spite of the requirements and remedial courses, many students still are afraid of symbols, do not understand what a formula tells them, and often cannot see past a particular application to the general concept it is intended to illustrate. Thus diversity in mathematical insight provides another major source of heterogeneity with which we must deal.

The appearance of various comprehensive statistical computing packages has led to a popular, but rarely stated fallacy: If students can learn to use a statistical computing package well, they do not need to take a statistics course. Nothing could be further from the truth. In order to effectively use a statistical computing package, the user must decide which program or procedure in the package fits his/her needs, and then choose appropriate options. In a particular case, there may be 30 to 100 programs or procedures to choose from, and each may have as many as 20 major options. Several of the major statistical computing packages are designed so the user has to employ several different procedures, in an specific order, to accomplish an analysis which is appropriate for a particular situation. Clearly, a user faces tens of thousands of possible configurations of analyses. After a particular analysis is done, what does the stack of computer output, perhaps several inches deep, mean in the context of the original problem? Informed decision making in this computational context requires a good understanding of the theory, **NOT** necessarily the mathematics, behind the various procedures. This presents a real instructional challenge when students arrive with limited overall mathematics preparation, and seemingly more interest in immediate applications than in concepts.

2. COMPUTER GENERATED ASSIGNMENTS

The first illustration of computer usage concerns the assigned lab or homework exercises. One way to partially deal with students' concern about how a topic relates to their discipline is for some of the exercises to have contexts which are specifically associated with their respective disciplines. Further, it would be helpful if each student's data were somehow unique. The latter characteristic raises the level of interaction among students from "What answer did you get?" to "How did you approach that problem?" This shifts discussion among students from numbers to concepts.

Faculty in the Department of Experimental Statistics have developed a computer system for creating such problems, and for preparing personalized assignments for each student. Necessarily, accompanying answer sheets must be produced for the instructor, and for subsequent use by the students. Once problems are developed and the instructor makes decisions about what subject matter areas - students from related majors - get what problems, an assignment can be setup, executed, and sent to a high speed printer in less than ten minutes. The development of individual problems takes from one to several hours, but once done, they can be improved and reused. Students provide a good source of relevant problems and are good critics of the wording of specific questions.

Completed problems are stored in four parallel computer files, one each for the problem text, the specification for the generation of the random data to be embedded in the text, the answer text, and the way to compute the answers to appear in the answer text. All four files are closely linked by how the objects are named in the respective files. The problem text carries substantial identification and keys which characterize several features of the problem, including its subject matter area and its statistical type. With experience, the wording of the stored problems has been improved, partly in response to student criticism.

Another file contains information about a specific class, including a class list which gives each student's name, major, and broad subject matter area. This file also contains information about each assignment which is to be passed out. A class assignment contains header information appropriate for that assignment, and a table which relates the students' subject matter areas and the assignment's problem numbers to the identifiers of stored problems. For example, in assignment four, assignment problem one could be stored problem 72 for all students. Assignment problem two might use stored problems 78, 79 and 102; 78 might be used for social science and business, 79 for agriculture and biology, and 102 for the remaining areas. There could be more assignment problems if appropriate.

When an assignment is run, the controlling computer program requests the class identifier and assignment number. From this it retrieves the class list and the specifications of the assignment. It then loops through the class list placing each student's name in the appropriate place in the assignment header. Using the student's subject matter information, it retrieves the appropriate stored problems, generates that student's random data, embeds the data in the text, and otherwise prepares that student's assignment for printing. After the assignments for all of the students are completed, the accompanying answer sheets are prepared in an analogous manner. Each student's data is generated a second time by resetting the beginning point for the random number generator. Alternatively, the answers can be evaluated at a later time than when the assignments are created. This feature can be useful when new stored problems are being prepared because the answer text and answer generator can be prepared several days later than the problem text and its generating specifications.

This system has been used for several years by several instructors in four classes. We presently are considering expanding its use to another large, multisection class.

In order to effectively use this system, we need to clearly keep the objectives of a stored problem in view, and use it at the appropriate time. Even though each use of a stored problem leads to new data, we still must be careful with problem files which students pass on from one semester to another. This particularly impacts on how answers texts are written. If discussion questions have detailed responses on the answer sheet, some students will give almost that identical answer in subsequent uses of that stored problem. Instead, the answer text can ask questions which will point the student toward the appropriate ideas. We had anticipated that individualized assignments would take longer to grade. Much to our surprise this doesn't seem to be the case.

Is this computer usage limited to statistics? We think not. In principal the ideas advanced here could be adapted to any area which assigns quantitative problems to students. The value of making problem texts and data characteristics which reflect students' disciplinary interests would depend on the subject being taught. The value of student-specific data should extend at least to some courses in accounting, chemistry, physics, engineering, and mathematics. In fact the system we use relies on statistics only in the precoded ways to compute answers.

At present, the system described in this section is implemented on the academic mainframe computer at New Mexico State University. It uses the text editor named PRESS and is implemented in the computer language named APL. Certain details of the system depend heavily on this environment. This system was started with a modest grant from our university for student programming support.

3. DECISION MAKING IN AN ADVANCED REGRESSION COURSE

The second illustration of computer usage was designed to help students learn to make decisions within the context of regression analyses. Some statistical analyses require relatively complex computations. To make such an analysis appropriate for a particular situation, the user must select many specific computational alternatives. A user of this analysis at another institution, and/or at another time probably will not have the same computational alternatives as are available on a particular campus at a particular time.

We have developed a regression analysis package which is instructionally, not production, oriented. It forces the user to make many decisions which might be made for him in more production-oriented environments. Further, so that a student is not overwhelmed by the number of alternatives - many not yet discussed in class - the instructor can release alternatives as students become ready for them. This system has evolved through four distinct editions in response to constructive criticism from students, and in response to changing computing environments.

The system has three distinct parts: preliminaries, analyses, and termination. The preliminaries begin by identifying the student, or other user for subsequent information needs, obtaining information about the user's terminal, and providing an opportunity for information about the interaction. Then the system presents messages from the instructor. Here the instructor may place information about common or individualized data sets, or minor changes in the system. The body of the analysis has two major parts: development and identification of the data set to be analyzed, and the actual analysis. The termination phase deals with one important matter before shutting itself off, namely, printing. Earlier editions of this system were designed for printing terminals; the interaction with the computer would provide the desired results. As printing terminals are slow, and subject to serious maintenance problems, most of them were replaced by video, or TV-like terminals several years ago. In order to execute the system at a video terminal, the entire conversation and results had to be captured, a major change between the third and fourth editions. During the termination phase, the user can choose to have the captured conversation printed at the central computing facility; otherwise it disappears.

This system utilizes menus, questions with yes/no answers, and questions with numeric or literal (character or letter) answers to communicate with the user. The menus can present up to ten alternatives; the user makes a menu selection by entering a number or depressing an associated function key if the terminal being used supports these. Menus are used to present major alternatives. The system obtains specific information by asking specific questions. The mechanism for accepting responses allows the imposition of various conditions on the user's response. For example, if the answer to a particular question needs to be an integer between one and seven, the user's response is checked for this condition; if the input does not meet the conditions, the user is asked to try again. When the computer requests a piece of information, the prompt or requesting message is taken from a file. This system allows the possibility of prompts coming from any of three parallel files. This, in turn, allows the designer to have long, very informative prompts in one file, while very brief, but equivalent prompts exist in another file. A user can change the file providing the prompts at any time by giving certain specific responses to a question. This allows the user to get brief prompts in familiar parts of the system, but more lengthy ones in new parts of the system. Users will encounter this situation because the instructor can prevent any menu entry from appearing until an appropriate time, such as after the topic associated with the menu entry has been discussed. This locking feature has another characteristic: A menu alternative may be prevented from appearing until the user has completed other, possibly prerequisite, menu entries. For example, a user cannot proceed to the analysis menu until a data set has been defined for subsequent use.

The data entry menu allows a user to retrieve his/her own data set previously generated by the system described in the preceding section, to enter data directly, to make a variety of data transformations, to display any currently defined data set, to identify a data set for subsequent analysis, or to proceed to analysis. Two of these options currently invoke other menus to accomplish their tasks.

The data analysis menu presents several preanalysis options for examining a data set or defining an analysis model, an analysis, and several postanalysis alternatives. An alternative for plotting any currently defined aspect of the data set is always available. A user can return to the data entry menu as another alternative which terminates data analysis menu. A user may select any available menu alternative or sequence of alternatives as frequently as desired; this is true for both the data entry and analysis menus, provided the prerequisite conditions have been met.

The system has evolved in response to various student suggestions and criticisms. Other instructors have used this system when they taught the course for which it was designed. All have suggested improvements, but one has been particularly helpful. Even though the system has evolved over nearly

15 years, and has been extensively "student proofed", occasional errors still remain. The language environment in which this application is written, APL, allows sophisticated error trapping. Consequently, an application designer can incorporate processes which allow the application to recover from errors which would cause termination of the system in other language environments. This protects the user from being particularly victimized by the language environment in which the application is developed. The user can even invoke this error recovery process with the word QUIT to escape from a menu alternative which he/she recognizes as inappropriate. The system records all errors which are trapped. The designer can examine these to identify and correct system features which fail under specific conditions.

This fourth edition is implemented on the academic mainframe computer at New Mexico State University. A fifth edition is partially developed. It is designed for use on microcomputers, IBM PC's in particular. Microcomputers provide several advantages over mainframes for this application. They allow much better control of screen displays, and thus a far superior menu driver. Further, the current campus computing environment is being expanded by the addition of a local area network for computer communication. When completed, this will allow network users to access and share files stored on any device accessible, with security, from the network. In particular, the developer can store an accessible version of the system on his microcomputer, and give his students access to it.

4. COMPUTER DRAWN TRANSPARENCIES

The third illustration of computer enhanced instruction concerns the development and use of overhead transparencies. These instructional tools solve some problems, but pose others. Instructors who use an overhead projector often use many handwritten transparencies due to the various "costs" of developing nicer ones. Furthermore, students often feel, legitimately, that instructors go through transparencies too fast. At present, we partially solve this problem by placing copies of handwritten transparencies on reserve in the library. Many students reproduce these, but complain, again legitimately, about the cost. An instructor rarely is willing to invest much more than this in handwritten transparencies.

We have tried to deal with most of these problems by developing appropriate computer software. By interfacing an inexpensive pen plotter with a desktop or microcomputer, the micro can drive the plotter to make transparencies. By implementing appropriate translation tables, the plotter can do special things such as change pen color, make subscripts and superscripts, or draw objects of special meaning in a discipline. Once setup, the micro can have the plotter make a transparency in five to ten minutes. This goes a long way toward solving the problem of making decent transparencies quickly and in a timely manner. It also allows simple editing and re-creation of transparencies which don't accomplish their objective well.

But what about the students? The micro can have the plotter reduce the transparency to half size in both height and width so that four fit on one side of a piece of paper, and four more on the other side. Most lectures use no more than eight transparencies; a copy of all eight can be given to each student inexpensively, on a single sheet of paper. A set of transparencies which accompanied the oral presentation of this paper appear after the references as an appendix.

This application has been implemented on an IBM PC with a Hewlett-Packard model 7475A graphics plotter. A microcomputer version of the editor named PRESS,also used for the application discussed in section 2, is used to create texts. This editor has been expanded for the present application by adding commands to translate certain character patterns into parts of formulas and statistical symbols. This application is developed, implemented, and in final testing stages. I anticipate using it this spring term.

This sort of use of microcomputers could become widespread. I know of a faculty member in another department who is using a different kind of computer in similarly to that described here. Currently he is using it more for scientific presentations than for teaching, but his tools would serve teaching equally well. Some microcomputers, such as the Macintosh, have features which are well suited for making some kinds of transparencies. The image of a desired transparency can be created directly on the screen, and then printed. Most copy machines available in universities will make transparencies from a printed original of this sort. Transparencies are being made this way.

5. COMPUTER GRAPHICS FOR STATISTICAL TESTING

The fourth illustration of computers to enhance instruction involves computer graphics; this application is in its development stages. It is mentioned here to emphasize the importance of planning so that the technology will support instruction, not become the object of interest.

The topic of statistical testing seems to be difficult for students to grasp. But grasp it they must; otherwise their use of statistics will remain rote, and will frequently lead to misinterpretations. We are beginning to plan dynamic computer graphic demonstrations as part of an instructional module on statistical testing.

The module has one overall objective: understanding the concept of a statistical test. Mastery of this overall concept requires an understanding of these components: a probability model; null hypotheses; alternate hypotheses; types of errors; impact of the null hypothesis on the probability model; impact of the alternate hypothesis on the probability model; a division of outcomes into two sets (acceptance region, critical region); probability of errors under the null and alternative hypotheses, respectively. All of the components will be discussed in lecture. Once students have begun to grasp the ideas of null and alternate hypotheses, and of types of errors, computer graphics has promise for clarifying the remaining ideas. In particular, the general probability model can be shown one place on the screen, how it becomes specific under the null and alternative hypotheses can be shown elsewhere. Students can be allowed to experiment with the division between the acceptance and critical regions. The appropriate parts of the respective distributions can be shaded, and probabilities of the errors shown in "thermometers". As students move the boundary, the probability "thermometers" will fill or empty. Students can have assignments to get regions with specified characteristics.

An important feature of this planned application will be to allow students to experiment and get nearly instant feedback. Presently such a experiment requires extensive use of tables. The logistics of using tables becomes such a hindrance to many students that the important concepts drown in table usage.

6. CONCLUDING COMMENTS

Does it take much time? Is it worth it? Emphatically YES, to both questions.

The use of computers as a teaching tool adds little, if any, time to the teaching of a course. The time goes into preparation of materials. Before you can begin to use a computing tool, YOU must become familiar with it! How much time this takes will depend on the computing tool you plan to use, and on the support structure you have at your disposal. Even if you are lucky and have support to develop an application, you have to understand the tools well enough to decide how they fit your teaching situation. If a student employee, or a technician is qualified to make the teaching decisions, why are you needed? Presumably, you have the knowledge and experience to know how to approach the subject matter in the context in which it is to be taught. Either you will have to develop the tool yourself, or you will have to direct the development by someone who knows the technology well.

Appropriate technology, appropriately adapted to your context, can greatly improve your communication as a teacher. It may open avenues so you can reach students you previously lost. It may provide totally new ways to explain a topic. You may see the topic in a wholly new way. You can save your successes, improve on your acceptable presentations, and discard your failures. Don't plan to always succeed. If your channels of communication with students are open, you will know when you have succeeded, and when you have failed.

Success will nearly always require planning what you are going to try to achieve. The instructional objectives should remain as the guiding force. For almost all of us, the computer should be viewed as a tool, not as the object of interest.

7. REFERENCES

Southward, G.M., N.S. Urquhart, and M. Ortiz (1983). Computerenriched instruction of intermediate level statistical methods. **PROCEEDINGS of the SECTION ON STATISTICAL EDUCATION, AMERICAN STATISTICAL ASSOCIATION.** 1983. pp 6-9.

8. APPENDIX

See the following three pages for illustrations of transparencies of the sort discussed in section 3. These accompanied the oral presentation.

211

TEACHING-1

SOME CURRENT AND PROJECTED USES OF COMPUTER TECHNOLOGY IN THE TEACHING OF STATISTICS

by

N. Scott Urquhart
New Mexico State University

OBJECTIVE: Suggest, by illustration, how computers might be used to enhance instruction

CONTEXT: Applied statistics; level: introductory to intermediate

ILLUSTRATIONS:

> Computer generated assignments

> Decision making in regression

> Computer drawn transparencies

> A use for computer graphics

TEACHING-2

INSTRUCTIONAL REALITIES IN TEACHING STATISTICS

STUDENTS DISPLAY *HETEROGENEITY*

- * Subject matter areas - majors
- * Motivation - relation to major
- * Mathematical preparation
  - symbol shock
  - meaning of formulas
- * Experience with data

SECTIONING ON MAJOR and/or PREREQUISITES

PROVIDES ONLY A PARTIAL SOLUTION

===> COMPUTERS CAN *INDIVIDUALIZE*

TEACHING-3

COMPUTER GENERATED PROBLEMS

OBJECTIVE: Produce personalized problem sets. Each student's problem set should have:

- * UNIQUE DATA
  - changes interaction among students
  - virtually eliminates copying
- * CONTENT WHICH REFLECTS STUDENT'S MAJOR
  - equivalent instructional content, regardless of the problem's disciplinary content
  - problems differ only in nouns, adjectives, and data size
- * MOTIVATIONAL VALUE
  - increases interest
  - "relevance"
  - unanticipated: student's *pride* in their *own* problems

TEACHING-4

PROBLEM FILES

PROBLEM TEXT
P0215

PROBLEM GENERATOR
PGP0215

ANSWER TEXT
ATP0215

ANSWER GENERATOR
AGP0215

TEACHING-5

COURSE FILES

CLASS LIST

Name	SS no.	Subject Matter Area

ASSIGNMENT 1

Assignment Header
(points, due date, etc.)

Subject Area	Assignment Problem Number 1 2 3 .....
1	
2	stored problem ID (P0215) (problem selection matrix)
:	
:	

ASSIGNMENT 2

:
:
:

ASSIGNMENT N

TEACHING-6

RESULT

Assignment Number Name: *Your Name*
Other Information
(Ex: Date Due, value, etc.)

PROBLEM 1
- Text with interspersed numbers
- Text and format specifications from problem text file
- Numbers randomly generated according to specifications in problem generator file
- Problem text used depends on your discipline area

PROBLEM 2

:
:

TEACHING-7

DECISION MAKING IN REGRESSION

LEVEL: Intermediate Graduate, Nonmajors

OBJECTIVES:
* Comprehensive regression
* Easy to use
* Decision oriented
* Instructionally oriented
* Instructor controllable

DEVELOPMENT: Off and on since 1971

PRESENTLY:
* Menu driven - states alternatives
* Menu selections produce conversations, mainly yes/no
* User can alter verbosity of conversation
* Instructor controls availability of menu entries
* Randomly generated data retrievable

TEACHING-8

COMPUTER DRAWN TRANSPARENCIES

> TRANSPARENCIES CAN HELP LEARNING
- Focus attention
- Allows instructor to accurately record details
- Can provide a record for absent students
- Keeps lecturer on intended track

> PROBLEMS
- Note taking difficult for students
- Lecturer may go too fast
- Handwritten ones may be hard to read

> SOLUTIONS
- Have computer draw sharp originals
- Reduce for students (computer can)

> ADVANTAGES
- Quicker than handwritten, or professional artist
- Storable for alteration

TEACHING-9

COMPUTER GRAPHICS
FOR
STATISTICAL TESTING

THIS IS AN IDEA UNDER DEVELOPMENT

OBJECTIVE: To provide a visual experience to complement usual lecture explanation

FEATURE: By providing an environment in which experimentation is easy and acceptable

HOPE: Some students who presently do not understand this idea may get a real second chance.

TEACHING-10

THE BOTTOM LINE

FRUITFUL USE OF COMPUTER ENRICHED INSTRUCTION REQUIRES:

- *OBJECTIVES*
- *PLANNING OF THE COMPUTER'S ROLE*
- *IMPLEMENTATION*
- *UTILIZATION OF STUDENT FEEDBACK*

===> IS IT EASY? *NO!*

===> IS IT QUICK? *NO!*

===> IS IT WORTH THE EFFORT? *YES!*

STUDENTS, LEARNING AND COMPUTERS:
SURVIVING THE CHANGES

by

James C. Wade

University of Arizona

STUDENTS, LEARNING AND COMPUTERS: SURVIVING THE CHANGES

by

James C. Wade*

Introduction

In the desert beside the massive Columbia River as it separates the states of Oregon and Washington, computers on large commercial farms daily call a computer at the local electric cooperative to determine the appropriate quantity of irrigation water to apply to the farms' crops. The quantities of water are optimized daily considering the age and type of crop, past irrigations and the recent local weather history. The cooperative receives satellite weather data periodically throughout the day from its solar powered weather station located among the farms. The data required for complex water management decisions are collected, analyzed and distributed on the cooperative's microcomputer and are processed on using similar computers. In Colorado and Nebraska, computer controlled irrigation management systems actually turn on and off the irrigation pumps to take advantage of time of day electricity pricing (Duke, et al., 1984).

These are not futurist views of agriculture. These and other similar applications of computers promise to create major changes in agriculture. American farmers have traditionally been rapid to adopt new production technology such as mechanization, hybrid seed and agricultural chemicals. Such adoptions have made American agriculture a model of efficiency (Thurow,

---

* James C. Wade is Associate Professor in the Department of Agricultural Economics, College of Agriculture, The University of Arizona, Tucson, Arizona. Paper No. 513, Arizona Agricultural Experiment Station.

1983). Sonka (1983) identifies three major revolutions in agriculture: substitution of animal power for human power, introduction of mechanical power and adoption of chemical-backed techniques, and emphasis on business and financial management for farm growth and/or survival (the information age in farming). Information based technological changes are already making significant changes in agriculture as the demand for important decision making information grows. Farmers are both consumers of outside information and suppliers of the internal information services required for effective business management. New agribusiness and government sectors are emerging to help provide computer and information services needed in agriculture. Knowledgeable and successful users of such information services increase their likelihood of survival in the competitive farm sector.

Although many anticipated computer innovations may be years away, recent changes are having major impacts on teaching. Educational institutions are pressed to produce students with at least the rudiments of computer knowledge, whether the student will work on the farm, in agribusiness or in government. Colleges of Agriculture, like other educational institutions, are searching for the best instructional uses of computers in their varied undergraduate and graduate programs. Rapid changes in the computing environment and insufficient experience and exposure of newly graduated students increase the urgency of the transition from traditional methods of teaching toward those that integrate computers into the overall teaching process. Students with better computer and information management skills increase their opportunities to survive and prosper in the food and fiber sector of the information age.

Economic survival in agriculture depends more and more on obtaining and successfully processing the information sufficient to assure a comparative production and marketing advantage. Table 1 lists some of the areas in which

Table 1. Information Environment in Agriculture

- Marketing Information
  - Output Price Access
  - Customer Access
  - Availability of Inputs
  - Input Prices
  - Weather Impacts on Markets
  - Market Strategies
  - News Related to Marketing and Production
  - Inventory and Storage Control
- Financial Management and Accounting
  - Record Keeping
  - Cash Flows
  - Cost Accounting
  - Cost Budgeting
  - Investment Analysis
  - Payroll
- Production Decision Making
  - Crop Mix
  - Input Timing
  - Production Risk
  - Feed Mix
- Extenstion Information
  - Technical Advising
  - Market Information
- Scientific Control of Complex Process
  - Milking
  - Irrigation
  - Feed Mixing
  - Information Handling
  - Input Monitoring

computers have been, or may be, of considerable benefit in farming. Other applications exist for the much larger agribusiness and related government sectors. It is readily apparent that most of these impacts are in the area of business management decision making, which directly impacts on the instructional programs of Departments of Agricultural Economics and Agricultural Business.

Providing agricultural students with a foothold in, and helping them become efficient participants in, the society is the goal of all instruction in Colleges of Agriculture. Increasingly, this goal requires knowledge of the functions and uses of information and computers in business management and economic policy analysis. The "information age" is upon University instruction.

The objective of this paper is to outline some approaches for using computers to aid in the teaching of students in Colleges of Agriculture and to share some ideas for improving instruction in the rapidly changing computer environment. The fundamental assumption of this paper is that computers are *not* the end but the means in educational computing. They are not a substitute for classroom teaching, but an aid in gathering and processing information that leads to effective knowledge of computers and how they may be used to solve problems, especially those of agricultural economics and agricultural business management.

This paper first outlines a conceptual relationship among students, learning and computers in university instruction. Next, the "ideals" and the "realities" of instructional computing are discussed. A prospective on moving from the reality toward the ideal is then developed from the author's Department's experiences in "muddling through" the changes. A summary and conclusion provides some keys to surviving the rapidly changing microcomputer instructional environment.

Students, Learning and Computers

The three most important aspects of computers in instruction are students, learning and computers. Bringing students to knowledge is the focus of all education. In this case, the goal is to integrate the use of computers into the education process.

Students

Students are, of course, the recipient of the instructional process and few university instructors misunderstand this idea. Students are why teachers and universities exist. But, to put students into the context of this discussion, some of their characteristics must be recognized. First, students in most academic areas have *not* come to the university primarily seeking an education in computer applications or computer science. Otherwise, they would have chosen those areas of specialization. Students in Colleges of

Agriculture, to some extent, are interested in the applied biological aspects of plant and animal production, in the technical systems used to manage these biological systems and/or the social and economic environment in which these processes are taking place. These students are not, and do not want to be, computer scientists. Students in these "non-computer" fields are, however, aware of the increased role of computers in the "information society" and that students in areas of study more directly related to computers (and engineering) have better job opportunities. Many of these same students have limited quantitative skills and, perhaps, even a bias or fear of computers and quantitative work. A few students view working at a keyboard degrading and mindless. Others recognize the benefits of knowing and applying computers in the chosen course of study but do not want to be burdened with the details of computer operations and programming. These students want and need to know how to use computers to supplement their applied discipline.

In summary, students are not as qualified as instructors might like but they are interested in learning to effectively use computers in their chosen area of study even though they may have no real idea what that learning might entail.

Learning

Having briefly examined students, instructors wishing to use computers in an educational program must examine their teaching goals. Once the goals are firmly established, means of achieving those goals may be defined tc include computers. Table 2 gives a partial list of the areas in which the computers may aid in university instruction.

In the disciplines underlying applied biological, physical and social science education, abstraction and theory of complex ideas play major roles in

Table 2. Roles of Information and Computing in Non-Computer Oriented Instruction

---

- Provide supplemental instruction (Tutorals)
- Provide supplemental drill
- Provide access to supplemental information
- Provide examples
- Provide easy anaysis of alternative solutions
- Solve complex problems with large computing requirements
- Knowledge of the role of information
- Experience with computing environment
- Testing
- Accurate visual presentation
- Hands-on experience

---

establishing a learning environment. Some such abstractions are never more than that to most students. While others may be developed into immediate and long lasting applications. Indeed, a common complaint of students is of a lack of usefulness of the ideas provided in, for example, economics classes. Clarification and utilization of abstraction and formal decision rules through examples, drills and exercises are, indeed, often a major portion of most educational programs. Applications have always been the key distinction in the linkage of complex ideas to agriculture since the goal is to efficiently produce the crops and animals required by a growing society. For this reason, agricultural education has always placed a strong emphasis on laboratories and field trips in its classes.

"Learning by doing" is one of the oldest maxims of education. The concept of applied/hands-on experience is a primary component of modern learning systems, especially in the sciences and agriculture. Laboratories have become an essential element in the continuing scientific revolution. The fundamentals of theoretical and conceptual analysis and the factual knowledge provided in classroom lectures notwithstanding, the practical experience of "doing" is essential to the precepts of beneficial education. The subject may be writing, mathematics or technology. Inexpensive computers and related electronic technologies have the potential to influence classroom applications by providing more opportunity to apply the knowledge learned from conceptual analysis.

Facts, another mainstay of education, if presented in a void of conceptual ideas, provide little opportunity for learning. Data may be increased, but useful information and knowledge may not be increased. Integration of concepts, theories, hypotheses, inquires and information are at the heart of modern learning. Separating, categorizing, and utilizing important information is essential in testing and applying conceptual ideas. The drudgery of such work is reduced by the supplemental use of computers.

Thus, a summary question is "What are the goals for using computers in instruction and do they blend into the overall objectives of learning?" Is the goal to supplement teacher and tutoral shortages by substituting computers? Is the goal to teach basic computer use and programming? Are computers to store and retrieve data and information? Are computers to be used to clarify abstract conceptual ideas through applications, demonstrations and examples? Some (e.g., Litzenberg, 1982) assert that computer programming is a prerequisite for computer based learning and that computer literacy requires a knowledge of how computers function and how software is designed. Are these requirements part of the goals of instruction in the parent discipline? And finally, can learning of the central elements of the discipline be improved using computers? If not, then compuers may not be needed in instruction.

All of the computer uses suggested in the previous paragraph are valid to some degree. However, in agricultural economics, the primary goal is to train knowledgeable decision analysts and decision makers, whether they are producers, agribusinesses, or resource or policy analysts. Computers are simply tools for aiding this process.

Computers

The evolution of computing hardware and software are the factors that give impetus to the urgency of the questions addressed in this paper. In less than five years, the computer industry has been revolutionized and the concept of personal computing has become a reality. Problems that only a few years ago would have required a large mainframe computer now can be solved efficiently using "desktop," or even "kneetop" computers. Computing power is in smaller, cheaper, and easier to use packages.

A software revolution is following the hardware revolution and software ultimately provides the key to educational use of computers. Applications software design has dramatically improved over the past year. Integrated word processing, spreadsheet and graphics systems are readily available (although not always inexpensive) for almost every make of microcomputers. Decision aids and information handling software for various applications, from accounting to statistical analysis, are expanding rapidly. As availability and competition expand, prices decrease and capabilities increase. These software aids, designed for efficiency by knowledgeable computer users, utilize the full power and promise of microcomputers by using efficient data handling, interactive computing and full screen graphic display. New software is designed to be as "transparent" as possible for the user. As a result, user burden is minimized.

Rapid change in computing hardware and software is the current and foreseeable state of the art. Hardware or software purchased today is superceded by more powerful and cheaper versions in just a few months. Stabilizing instructional computing in such an environment is, at best, difficult especially for a faculty that is primarily interested in the concepts and applications of their discipline. Generating new and truly unique computer applications for instruction is all but impossible. Just keeping up seems impossible. Many faculty simply will not bother.

The availability of powerful, inexpensive computing aids the underlying concept of the educational process by allowing students to have meaningful hands-on experience at a relatively low cost and with considerable ease. It is tremendously important that at least some instructors lead the way in developing solid educational applications for microcomputers in disciplines where they enhance existing educational goals. By briefly examining the "ideals" and the "realities" of microcomputing in an instructional environment, this paper seeks to enhance the understanding of the alternative uses of computers in instruction.

The Ideal Instructional Computing Environment

Proper integration of students, learning and computers should lead to an ideal instructional computing environment. The key elements of this environment for non-computer disciplines are outlined in Table 3. The ideal operating environment is largely transparent to the student and the instructor. In such cases, the student has a knowledge of the basic operational procedures of, and some experience on, the computers available. A previous fully integrated and coordinated course of studies has prepared the student to use microcomputers, at least to a limited extent. The student knows how to efficiently use the keyboard and has no inherent fear of computing. The student understands the basics of mathematics and equation

Table 3. Ideals for a Computer Instruction Environment

Student Literacy
- Knowledge of the Basics

Well Integrated Educational Goals
- Administrative Support
- Conceptual Integration
  - Innovative and coordinated goals
- Knowledgeable Administrators
- Support Personnel, Supplies and Maintenance
- Software
  - Innovative and coordinated to goals

Student Access
- Hardware Availability
  - Compatible and inexpensive
- Small Groups
- Software Availability
  - Compatible and inexpensive

Well Structured Software
- Drills and exercises
- Tutorals
  - Computer Literacy
  - Concepts and Theories
  - Introduction to Software
- Problem Solving
  - Statistics
  - Spreadsheet
  - Operations Research
  - Financial Analysis
  - Accounting

Instructors
- Know Concepts
- Know Software Use

structure. In short, basic computer literacy already exists so that the operational aspects of computing need not be retaught. All students are be more or less at the same level of knowledge.

The ideal computer facility, or laboratory, is supplied with up-to-date microcomputers including, at least, a printer and for each computer, graphic video output. Memory and data storage devices are adequate for the largest software requirements. Each machine has a telecommunications access to data utilities and other outside larger computers. A sufficient number of identical computers are available. Few, if any, students are waiting to use the computers. The laboratory is spacious and pleasant. Machines and students are placed so that distractions are at a minimum. Conflicts of time and space do not exist.

In an ideal instructional environment, administrators at all levels, from Department Heads to the Legislators, are knowledgeable of, and appreciate the use of, computers as intructional tools. Adequate support personnel, supplies and maintenance are provided. Instructors are rewarded for innovative teaching in a well understood and coordinated way. No conflicts exist as to which departments are to teach which materials and maintain computer laboratories. Every department that wants a laboratory has unlimited access to one.

Software, the brains in the ideal computer laboratory, is easily chosen from that readily available in the software market and funding exists to purchase, in large quantities, the exact software needed for each instructional goal. If not available on the market, an expert programming staff is available to design the needed software. Drills, exercises, tutorals, and problem solving units are precise and serve exact educational needs. Instructors are fully knowledgeable, do not make mistakes, and integrate the software presentation into other material given the student.

The "Ideal Instructional Computing Environment" is, of course, only a goal. Perhaps, the "ideal" presents nothing more than the way changes may ultimately provide for future instructional computer use. In the meantime, instruction is carried out in a more "realistic" environment.

The Real Instructional Computing Environment

The "real" instructional computing environment is much the opposite of the "ideal" presented in the previous section. Demands from students, parents and administrators seek increased and improved computer skills without providing the resources needed to meet these demands. Like other laboratory sciences, the benefit of computers in instruction is in the hands-on experiences of the students. However, computer laboratories for instruction are assumed by some administrators to be necessary for "computer sciences" and "engineering", but not for other academic areas. Such laboratories, if ever built, are assumed to retain their usefulness almost indefinitely and are costless to maintain. The rapid change in software and hardware is not fully acknowledged. Computing laboratories are expensive to establish, supervise and maintain. Space is usually limited.

Thus, the instructional laboratory in a "non-computer" discipline, in reality, is a few (if more than one) microcomputers or mainframe terminals in a small unused storage space or in a faculty member's office. Cramped conditions endanger equipment and limit learning potential. Limited equipment create over use further shortening the useful life of the equipment. Graphics presentation may not exist, or is very limited, and printers are often inconveniently shared. Telecommunications is restricted or nonexistent.

Many students enter courses that use microcomputers with little or no previous training. The instructor teaches "computer literacy" and, thus, removes a bit of material from the original planned course outline. Many instructors find the benefits of computer base instruction large enough to warrant the rescheduling of other activities. Instructors often lack knowledge of the potential uses of computers in instruction, of the method of coordination of computer aids into their instructional goals and of the available software for their application. A majority of instructors could not, or would not, write software for such computer based applications. Knowledgeable programming or teaching assistance is rarely available. Little or no academic recognition is provided for innovative teaching and software design in most large universities.

Software cost is high, especially if several different software packages are being used in a computer laboratory. Of course, software packages are not directly available for many conceptual ideas of primary concern. Software application is the central focus of instructional computing. If instructors are waiting for an ideal software package to aid the learning of difficult concepts, the wait may be quite long. In fact, the instructor may need to write, or hire a programmer to write, a program if the software need is ever to be fulfilled. Funds for such activities are limited.

Although limitations on software for university level instructional use appear to be very severe, these limitations are primarily in the areas of preprogrammed tutorals, drills and examples. Software markets have greatly improved for business and accounting, statistical and operations research software. These, usually expensive, softwares are often discounted in volume to educational users and provide an ideal base from which to develop the types of educational application that best fit many university instructional environments.

Given the realities of the instructional computing environment, what approaches are appropriate and how can they be implemented? To illustrate one approach, the following section describes the recent efforts in the Department of Agricultural Economics at the University of Arizona to utilize computers as a component of instruction. Efforts that are less than perfect but efforts that provide a beginning.

Muddling Through: A Case Study

To place some prospective on the potentials of and approaches to microcomputer in instruction, the Department of Agricultural Economics at the University of Arizona is used as an example (see Wilson, 1984). The Department has approximately 28 microcomputers to service 16 full-time faculty plus support staff in teaching, research and extension. Microcomputers are an important element of office productivity and the research and extension programs of the Department. Additionally, eight of the 8-bit CP/M based microcomputers are dedicated to instructional use.

The Department's 22 courses are taught once each year. Class sizes are relatively small, ranging up to approximately 60 students. Twelve faculty positions provide full-time equivalent instructional faculty of approximately 3.12 faculty years/year. The Department has one half-time teaching assistant who works primarily with computer oriented courses. The Department has 27 graduate students. The Department services some instructional needs for other departments of the University. Instructional computer use is carefully limited to a role of supplementing existing courses in Farm Management, Financial Analysis, Production Economics, Agricultural Marketing and Demand Analysis, and Applied Operations Research.

The last two of the courses listed above are graduate courses and reflect the traditional linkage of computers to applied research in agricultural economics dating back some 20 or more years. Graduate instruction traditionally includes at least some instruction in, and experience with, computer software and, to a lesser extent, computer programming. Particular applications center on statistical analysis (including econometrics) and mathematical programming. At the University of Arizona, graduate class sizes are relatively small (10 to 30 students) providing for significant individual instruction.

The traditional approach to graduate instruction has been to assign students small to moderate size problems for analysis. The students are given minimal instructions of data entry and formating and are left to determine how the problem is to be solved using the assigned software. Students must setup the problem properly, convert the problem to the nomenclature of the available software and produce and analyze the solution. These computer related assignments have primarily been in the areas of mathematical programming. The software interfaces have been primarily through batch card entry since student access through timesharing is somewhat limited and cumbersome. In recent years, some of the data management and other functions of instructional applications have been shifted to timeshare/ interactive computing on moderate to large computers which serve as "frontend processors" for the large mainframe computers. The students correctly perceive this approach as antiquated and resist learning such processes. Many have never learned to use a card punch or card reader and see little need in doing it now. Experiments using timeshare systems as frontend processors for the batch input have proven less than satisfactory since few students have experience with line oriented text editors.

Although this traditional approach is limited, substantial learning in problem applications is accomplished. The transition from batch to timeshare to microcomputers is underway but limited by available applications software and timeshare access. It seems appropriate to jump over this stage and move directly to microcomputing as a substitute.

Formal microcomputer use in graduate courses has, thus far, been limited to a few test cases in linear programming and statistical analysis. However, for many applications at the graduate level, efficient software for microcomputers is just now becoming available. As microcomputer/software capabilities improve, all but the largest statistical and operations research applications will be solved using microcomputers.

Graduate student learning on microcomputers is a more informal process. Most graduate students see immediate applications of microcomputers for word processing of papers and theses, and some for other applications. The microcomputer laboratory (including some additional machines dedicated to research) is made available to graduate students without supervision on essentially an unlimited basis. Thus, students recognizing the potential cost savings of the word processing teach themselves. Ultimately, they learn software necessary and valuable for carrying out thesis research. A few even learn to program in the languages available to them, although learning to program is not usually necessary. This informal learning is an important aspect of graduate training.

Computer applications in undergraduate instruction have historically been less because of software complexity. Some of the applications software used in graduate courses and in research has been used with limited success at the undergraduate level. Because of required background and experience to run large mainframe software packages, these software applications at the undergraduate level have been met with mixed results. The applications and software are simply more advanced than needed at the undergraduate level. Microcomputers are changing this complexity and providing significant opportunity for increased instructional applications.

In the Fall of 1982, the first attempts were made to use microcomputers as an aid in undergraduate teaching after the Department received a small gift from CALCOT-Seitz Foundation to purchase a few portable microcomputers for instruction. Although the computers arrived later in the semester than was desirable, some applications were developed and successfully carried out in this first semester. Several factors, besides the late arrival of the computers, affected the first attempt to use the microcomputers to supplement classroom instruction. First, since adequate space was not available to establish a student laboratory, the instructors attempted to setup the microcomputers during regularly scheduled classes. The computers had to be removed immediately following the class. This idea seemed reasonable since the computers were portable. For introductory instruction, this arrangement was adequate, although additional personnel were required for equipment setup and monitoring. The most limiting part of move-in/move-out instruction is that it does not provide "out-of-class" access to the computers so that each student may spend some hands-on time learning the functions of each key and the subtleties of the software and to experiment with applications.

Although student enthusiasm was high and some learning of microsystem and software applications did occur, the overall experience called for a change. For the second and third year, a small microcomputer laboratory has been established which provides students with substantial time (7 hours per day) for experimentation and applications. Although the laboratory is cramped and access limited to eight microcomputers and two printers, this laboratory is proving a popular addition to the Department's instructional program.

The instructional goals and the approach to computer use in the Department are to use microcomputers and, with preprogrammed software, to aide in solving problems of interest in Agricultural Economics. The computers are used as laboratory tools and their use is flexible within the context of available software. Since emphasis is placed on applications, minimum start-up instruction is required and students can be involved in applications of their own within minutes of first coming in contact with the computer. To break the computer ice and to put the students at ease at the keyboard, initial applications are to type a business letter or research report using word processing software. A tutoral program is available on the word processor and students are given an instruction sheet with which to get started. Each student is required to provide their own applications diskette. Perfection is rarely obtained on the first exercise. But, as experience and acceptance increase, class assignments improve in quality of presentation and analysis. Additional assignments in the undergraduate courses include designing spreadsheets to coordinate data and to analyze farm management problems such as farm enterprise budgeting, cash flow analysis and capital investment analysis. Students are not given spreadsheet templates into which they simply plug numbers. If they are to successfully apply microcompuers when they leave the University, they must know the basics of software structure and applications. Computer programming is not taught but software applications are taught within the full context of the software. This approach, although more difficult for instructors, is proving beneficial for students.

In more abstract courses, such as Production Economics, spreadsheet analysis is used to augment the learning of decision processes involving production functions, budgeting and time value of money. In addition to spreadsheet software, applications are also developed on preprogrammed software for investment analysis and linear programming.

The experiences of the Department and the faculty most directly involved in computer use in instruction have been mixed. The faculty sees good potential for improved student learning of fundamental concepts. The transition from mainframe to microcomputer has *not* been easy and is far from complete. As the faculty learns the benefits of microcomputing, objectives must be clarified, capable teaching assistants must be acquired and software must be evaluated, purchased and applied.

Keys to Survival

In summary, microcomputers for instructional use are still primarily a *promise* in most university applications. Junior and Senior High Schools are often better equiped for providing microcomputer instruction than are universities. Perhaps much of the basic instruction in computer literacy should be left to secondary schools and community colleges. However, university instruction can profitably use microcomputers to enhance the learning experience by using them as tools to aid students learn applications of primary concepts. The *reality* of resource shortages, indifference and resistence to rapid change can be overcome by individual departments and instructors who struggle to survive in the microcomputer instruction world.

Some keys to surviving the changes are briefly outlined below:

1. Know your students and your instructional goals. For educational goals, like all other goals, getting there is more difficult if you don't know where you are going or the vehicle which is to be used. Understand student needs and wants. But don't compromise on sound instruction of basic disciplinary knowledge.

2. Know and use your computers. Instructors must know their equipment and its uses (including software) to be able to recognize new opportunities for application. Those who know are better able to teach. Teaching by example is easy in computing. Be involved in all aspects of the computing exercise.

3. Use student-friendly applications software. Learning concepts is much easier when students do not have to overcome a lot of user burden to apply the software. Emphasize applications software that is easily applied to the overall instructional goals. Don't teach computer programming. Do provide the students with help.

4. Keep it simple but push the student. Place problem solving and concept applications as near as possible in a real world context. Students working alone may have difficulty handling the data and calculations. But, students, microcomputers and software can handle a lot more data and calculations. Students' use of the software should be simplified, but not below the student's capabilities.

5. Let the students do the work. Some students learn faster than others and sometime equipment, software and students take some time to ripen. Be patient, flexible and let the students do the work. Let students use software to create their own spreadsheets and solution files for problem analysis. Make the students work for answers. In this work learning occurs.

6. Don't lie and be willing to admit a mistake. Don't oversell students on the computer's capabilities. They will only be disappointed. Let them know early that computing takes work, but that it usually is rewarding. If you make a mistake, admit it. Neither computers, students nor instructors are perfect.

7. Know the limitations. Not every computer can do the same tasks, neither can software or instructors. Not every application should be solved on a computer. Don't push beyond the limits. Don't write software, use software.

8. Always look for something new and upgrade. Keep checking with friends, colleagues, dealers, journals and magazines for new applications for software and hardware. Search out the alternatives. Do example problems and experiment. Be ready to change.

9. Find the money somewhere. Microcomputing is expensive and the funds are not always available in standard instructional budgets. However, other sources of funds or products exist. Hardware and software producers often reduce the price or give equipment to universities, especially for classroom users. They fully recognize the market enhancement of educated computer users. If computer instruction is an important part of the educational goals, push for the resources needed to enhance the goals.

Change is reality. Information, knowledge and learning help the world deal with change. The universities' goals are to improve the understanding and application of knowledge to the realities of the world. Computers in instruction can make this process much easier.

Students who have learned their lessons are able to leave the university and use the microcomputer tools available to help solve the decision problems facing them in business, industry or government. A computer is but an aid. Basic knowledge must be instilled in every student, whether or not computers are used in instruction. The computer must never become the end.

References

Duke, Harold R., Dale F. Heermann and Michael C. Blue, "Computer Control of Irrigation for Electrical Load Management," Transactions of the ASAC, Vol. 27, No. 2, pp. 597-602, 608, 1984.

Litzenberg, Kerry K., "Computer Use in the Agricultural Economics Classroom," American Journal of Agricultural Economics, Vol. 64, No. 5, December 1982, pp. 970-977.

Sonka, Steven T., Computers in Farming: Selection and Use, McGraw-Hill Book Co., New York, 1983.

Thurow, Lester C., "Farms: A Policy Success," Newsweek, New York, May 16, 1983.

Wilson, Paul N., "Department-Based Computer Instruction: Experiences for Agricultural Economics," mimeograph, Department of Agricultural Economics, The University of Arizona, April 1984.

COMPUTER AIDED INSTRUCTION FOR FRESHMAN CHEMISTRY AT THE U.S. AIR FORCE ACADEMY

by

James R. Wright
Larry D. Strawser
John R. Balyeat
Larry P. Davis
Jesse L. James
Phillip Jung
Jonathan C. Noetzel
Donn M. Storch

United States Air Force Academy

236

COMPUTER AIDED INSTRUCTION FOR FRESHMAN CHEMISTRY AT THE U.S. AIR FORCE ACADEMY.

James R. Wright, Larry D. Strawser, John R. Balyeat, Larry P. Davis, Jesse L. James, Phillip Jung, Jonathan C. Noetzel, Donn M. Storch. USAF Academy, Department of Chemistry, USAF Academy, Colorado Springs, CO 80840-5791.

This paper presents the computer aided instruction (CAI) system used for freshman chemistry at the USAF Academy. This CAI system utilized four approaches to help the student master the fundamentals of our first-year chemistry course: (1) drill/practice/tutorial, (2) laboratory calculations, (3) laboratory simulations, and (4) "games". Our drill/practice/tutorial programs cover most topics taught in our course. The laboratory calculation programs help the student understand the experiments he/she performs in the laboratory. The laboratory simulation programs are used to "run" an experiment, a pH titration, for example, at a computer terminal. We use games to stimulate and maintain our students' interest while they learn chemistry. These games include a real-time, 10-level "dungeon and dragons" adventure game named ALCHEMY. The student's key to survival in ALCHEMY, of course, is knowledge in freshman chemistry.

During the past 4 years, the Department of Chemistry at the Air Force Academy has developed an extensive system of computer aided instruction (CAI) for students and computer management tools for our instructors. Our focus has been on developing programs for the freshman class in general chemistry. Every student enrolled in our freshman course is given an account on the Academy's general purpose computer system (VAX 11/780). A student who logs into the system will be offered a menu of choices, one of which is Chemistry CAI. When the student choses Chemistry CAI, he is presented with a list of chemistry related programs. These programs are designed to help the student in four general areas.

- Drill and tutorial of basic concepts
- Laboratory experiment calculations
- Laboratory simulation
- Chemistry games

We have many programs which provide tutorial instruction of basic freshman concepts such as stoichiometry, electronic configuration and gas laws. In addition to tutorial instruction, students may choose to receive questions from the computer to practice what they've just learned.

We also have a series of CAI programs which help the student perform calculations required for their laboratory experiments. Each of these programs present the student with a replica of the laboratory data sheet which the student completes. Given the student's data, the computer calculates the correct answer. The student can then check to see if he performed the calculation correctly. If he did not, these CAI programs lead the student step-by-step through the correct calculation method.

We also have programs which simulate laboratory experiments. Our most successful venture into this area so far is a simulation of an acid/base titration experiment using a pH meter. Since we must teach about 200 students during a single time period, a titration experiment involving the use of a pH meter represents a large investment in equipment. Usually, this leads to the use of phenolphthalein as an indicator to find the equivalence point during the titration, rather then using the pH meter. Thus, we have the need to provide the hands-on titration experience, but we don't have enough equipment to do so. We feel a computer simulation is a good compromise. This particular graphics program is run on color TERAK terminal interfaced to the VAX 11/780.

A student can successfully simulate the pH titration experiment on the computer. He has available all the normal glassware and solutions he would have in the laboratory. The computer selects an unknown acid and from the titration curve, he can calculate the ionization constant (pKa). We have several options we can employ depending on the situation. The program can give the student a standardized sodium hydroxide solution or force him to standardize with potassium hydrogen phthalate. The acid can be pre-selected or it can be an unknown acid. The program itself is written in a user friendly manner so that the student does not have to learn a set of syntax rules to communicate with the computer. It also does the necessary bookkeeping for the instructor.

We also have a few chemistry games designed to help the student learn chemistry while he is having fun. One of these programs is a nomenclature game called, appropriately, NAME GAME. Students are given 3 minutes to correctly answer as many questions as they can. Points are awarded or deducted accordingly.

Another popular game we utilize to reinforce basic concepts is ALCHEMY, a dungeons and dragons adventure type game. Unlike most adventure games, however, ALCHEMY does not require a player to learn his way around a maze, only that he correctly answer chemistry related questions when asked by one of Alchemy's resident creatures. If a student misses a question, the creature always gives him a reference to look up the answer.

Students are told that CAI is a tool designed to help them learn.

Hopefully, it will be fun and challenging. Students are reminded to use it as any other study tool. They are encouraged to use it only if they find it helpful.

Our CAI is used extensively by our students. During the first 4 weeks of this fall semester, there were over 22,000 separate program runs. The total program run time exceeded 1,700 hours.

The Department of Chemistry also uses the computer as a management tool. We use the computer to store test questions, to automatically generate tests and to manage our grades. For example, we have several data banks of test questions (PDP 11/70). This data base not only includes the test questions, but also includes the selectivity and ease index of the questions, key subject words, and source of the questions. We've written a program which can utilize these data banks to automatically generate the test in the proper format. Test questions may be selected manually or automatically by subject. This has the potential of saving our department a great deal of typing time. We also have a very large data base that manages the students grades (VAX 11/780). Instructors can easily and quickly store, retreive and print student information on a course section or individual student basis.

One other instructor program that is a tremendous time saver is our LABEVAL program (Burroughs 6900). When the students finish a laboratory experiment, they code their data on an optical scanner sheet. The computer reads these and performs appropriate calculations (usually includes per cent error) using the student's data. Hardcopy output of data and results are provided to the instructors and, eventually, the students. In checking the students' lab report, the instructor can very quickly see if the calculations were performed correctly. If they were not, the error is easily pinpointed.

Finally, a word on the Academy's Microcomputer in the Dorm or MID project. The Academy's goal is to have a microcomputer in every cadet room by 1987. These microcomputers will be networked to one common computer system. They'll be used as word processors, as CAI delivery systems, and as communication links between students and instructors. The specific microcomputer has not been selected at this time.